The Dissertation Committee for Wonhui Cho
certifies that this is the approved version of the following dissertation:

Temperature Control and Modeling of

Rapid Thermal Processing

Committee:

Thomas F. Edgar, Supervisor

David M. Himmelblau

Sanjay K. Banerjee

Si Z. Qin

Gyeong S. Hwang

Temperature Control and Modeling of Rapid Thermal Processing

by

Wonhui Cho, B.S. Ch.E., M.S. Ch.E., M.S. Physics

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2005

Dedicated to my family.

Acknowledgments

I deeply appreciate my supervisor, Dr. Thomas F. Edgar, for allowing me to work on experimental research. This research allowed me to apply knowledge on control theory I learned from classes and to become acquainted with vacuum systems, sensor technologies, PLC and Windows programming etc. Dr. Edgar's guidance on this research was clear and supportive in every aspect. I am also grateful to Dr. David M. Himmelblau, Dr. Sanjay K. Banerjee, Dr. Si Z. Qin and Dr. Gyeong S. Hwang for serving on my dissertation committee. I am also thankful to Dr. James B. Rawlings and Dr. I. Trachtenberg for initially serving as committee members.

I also appreciate the machine shop staffs in the Departments of Chemical Engineering, Mechanical Engineering and Chemistry for helping me building various attachments to my experimental setup. I am grateful to Mr. Robert Lewandowski of Chemistry Glass Blow Shop for his help with plasma tube. I would like to express special thanks to Mr. Eddie Oliver of Chemical Engineering for all the advice and help during whole time period of my experimental research.

I appreciate Dr. Jietae Lee of Kyungpook National University, Korea, for being my mentor of process control starting in my graduate school days back in KAIST. I appreciate Dr. Yoonsuck Choe of Texas A & M University for being my resource for computer software. His help with data acquisition programming is

deeply appreciated.

I am grateful to past and current graduate students of Chemical Engineering including my past officemates, Dr. James R. Bosley, Ron Ranly, Quay Finefrock and Dr. Tyler Soderstrom for sharing both joyable and hard times. I am also grateful to Dr. Anthony Toprac for his help with my experimental setup. And I also appreciate Dr. Keith Edwards for his help with various projects at Momentum Technical Consulting.

I appreciate many Korean graduate students who shared happy and difficult moments with me during this research. Thank you Won Sun Ryoo for your help with various Korean society activities. I am also indebted to Eui Kyoon Kim , Tae Sang Kim, Doh Chang Lee in Chemical Engineering and many other Korean friends in Austin for their nice support and friendship.

I am also grateful for my family's help during this research. I am deeply indebted to my parents for their financial and spiritual support while I was pursuing my PhD. I am very thankful to my mother-in-law and to my father-in-law, the deceased Korean Army General, Chin, Yong Gon, for their loving support.

I appreciate my children, Sherry and Peter, for understanding during all the hard times. I am extremely proud of both of them.

Finally, my deepest appreciation goes to my lovely wife Hyung Joo, without whose help, this work would not have been completed.

Temperature Control and Modeling of Rapid Thermal Processing

Publication No. _____

Wonhui Cho, Ph.D.
The University of Texas at Austin, 2005

Supervisor: Thomas F. Edgar

A Rapid Thermal Processing (RTP) chamber capable of plasma-enhanced CVD was built with associated computer control system. The RTP chamber can handle up to a 6 inch wafer and is equipped with 37 tungsten halogen lamps positioned in three concentric zones to radiatively heat the wafer. Thermocouple instrumented wafers with K-type thermocouple junctions embedded at the proper temperature measurement positions acquired the temperature responses from center, middle and edge regions of the wafer. Two narrow band gap infrared pyrometers ($\lambda = 3.3\mu m$) were installed at the top of the lamp housing with water circulating light pipes accessing the wafer temperature signal points. In RTP, accurate temperature control during ramp and hold with fast ramp rate and good uniformity is necessary. The RTP chamber showed ill-conditioned behavior for a 3 x 3 lamp power and wafer temperature multivariable system, showing relatively large interactions between controlled variables, which yielded significant temperature differences across the wafer. A baffle ring was tested to obtain an improved condition number and uniformity of

$\pm 1^{\circ}$C among three measured temperatures. Wafer effective emissivity and view factors were estimated from closed-loop identification of process parameters. Through the on-line estimation of the above parameters and simple closed-loop identification of the process parameters, such as time constants and process gains, one can detect wafer temperature variation from recipe set values when there is a unwanted deposition at the measurement side of the wafer. To accurately identify the ill-conditioned system, an iterative identification method was developed and effects of element uncertainties to control an ill-conditioned system was investigated. Furthermore, a multiloop control strategy was also developed. Finally, an iterative technique was introduced that allows a recipe-driven SISO(Single Input Single output) controlled system to automatically optimize the process recipe for optimal power ratios of each lamp zone.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Single wafer processing (SWP) is becoming popular in newer semiconductor FABs due to reduced production and reduced development(learning) cycle times. In 1993, Texas Instrument demonstrated single wafer integrated manufacturing (based on 150 mm wafer) under MMST (Microelectronics Manufacturing Science and Technology) program sponsored by the Department of Defense (Moslehi [62], Davis [20]). One of the key contributions of the MMST program was to introduce new concepts to semiconductor industry including real time control. Among SWP tools, RTP(Rapid Thermal Processing) is one of the first unit operations introduced in the 1980s (Singh and Nulman[83]). Recently, Edgar et al.[24] published a comprehensive review article on applications of advanced process control (APC) in wafer processing including RTP tools. Numerous research articles for single-wafer manufacturing technology have been published recently including application of RTP for manufacturing of advanced flash memory (Chen et al.[15]), batch(furnace) and SWP(RTP) process comparison in memory production by Weimer et al.[104] and feasibility study of RTP as an alternative to batch furnace for 0.15 μm 128-Mb mask read only memory(MROM) production (Hsu et al.[40]).

1.1 Rapid Thermal Processing

RTP has advantages over conventional furnace technology of operating with low thermal mass and rapid heating rates. RTP is widely used in metal silicide formation, implant annealing and gate oxide formation.

The least restrictive of these processes is perhaps the metal silicide formation, where a metal film deposited on the wafer surface is annealed with silicon to form a metal silicide (e.g., titanium silicide, cobalt silicide and NiSi). Titanium silicide has given way to cobalt silicide($CoSi_2$) because $CoSi_2$ offers lower contact resistance. And recently NiSi is being considered for ultra shallow junctions. Currently RTP is widely used in these applications because of its comparatively low thermal budget, its capability to purge undesirable gases (e.g., oxygen) before the wafer is heated.

RTP is also widely used for junction activation of semiconductor device fabrication. A MOSFET depends on two p-n junctions: between the source and the channel and between the channel and the drain. The sheet resistance of these source and drain regions should be as low as possible and the size of these regions are becoming smaller as device sizes shrink. In order for these regions to flow currents, the dopant ions should be placed in a substitutional locations in Si lattice. Dopant activation can be achieved at high temperature, and it does not need long time to activate the dopants. However, diffusion of the dopants during high temperature process would increase the junction depth. Increasing junction depth will fail the device. For both dopant activation mentioned above and implant annealing, where the lattice damages caused by bombarded high energy ions are repaired by thermal annealing, modern devices under critical junction depth control demands accurate

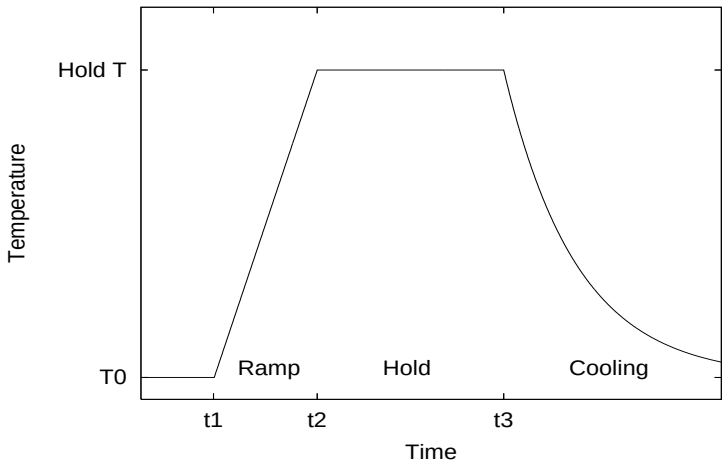


Figure 1.1: Schematic of RTP temperature profile

temperature control throughout the ramp and hold sequence of RTP (see Figure 1.1). In the implant anneal, a common specification calls for 10 to 30 seconds hold at temperatures between 950°C and 1050°C . Stringent temperature uniformity control is necessary to have consistent doping profile across wafer.

Due to its capability of fast *gas switching* and *thermal switching*, RTP can be widely used for thin gate oxide formation. Since gate oxide growth step precedes junction formation step, conventional batch furnace is still widely used due to its excellent wafer to wafer and within wafer uniformity in grown gate oxide thickness. But as the thickness of the gate oxide gets thinner, RTP can be widely used for its capability of stringent oxide thickness control.

Advanced CMOS devices will require junction depths between 13 to 22 nm in the source and drain region by the year 2005 according to the 2001 International Technology Road map for Semiconductors. Enhancement of RTP temperature con-

trol technology to keep the requirements of shallow junction depth specification is necessary.

Clearly instrumentation and control play a significant role in the operation of RTP systems. The developments of mathematical models either off-line or on-line and the determination of controllers between these models can provide improved controls of RTP, hence these topics are the focus of this dissertation.

1.2 Closed-loop Identification

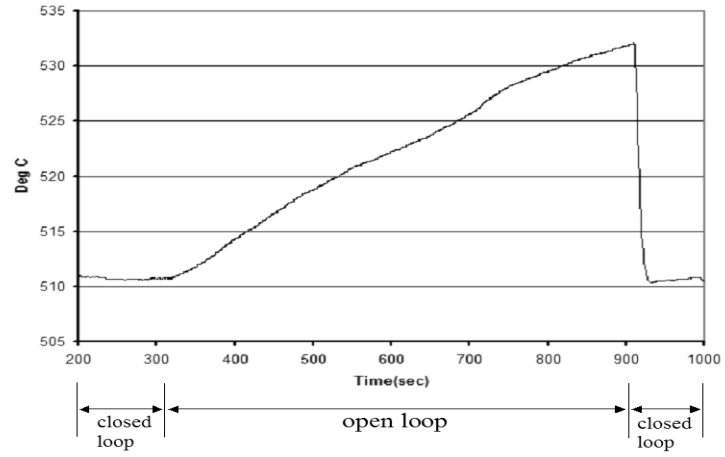
In an RTP chamber, temperature drift (in the absence of feedback control) can be relatively significant due to the continuous heating of the quartz window and surrounding chamber material together with insufficient cooling of the wafer. In Figure 1.2(a), a drift in temperature from about 500°C is shown. Initially, the wafer temperature was under feedback control with set point 500°C . At around 300 seconds, the PI controller was set to manual mode, leaving the lamp power levels at constant values (no feedback signal to the lamp power). Temperature increase of about 20°C was observed over 600 seconds without feedback control.

Identification of heat transport phenomena, i.e, identification of response characteristics of wafer temperatures due to lamp power changes for proper design of temperature control system may be affected by the drift mentioned above. For example, in open-loop identification, process gain and time constant values are estimated by detrending the drift from response curve. The estimation can be affected by the set point ramp rate and duration of identification time. This effect and the system nonlinearity will influence the design of the temperature control system.

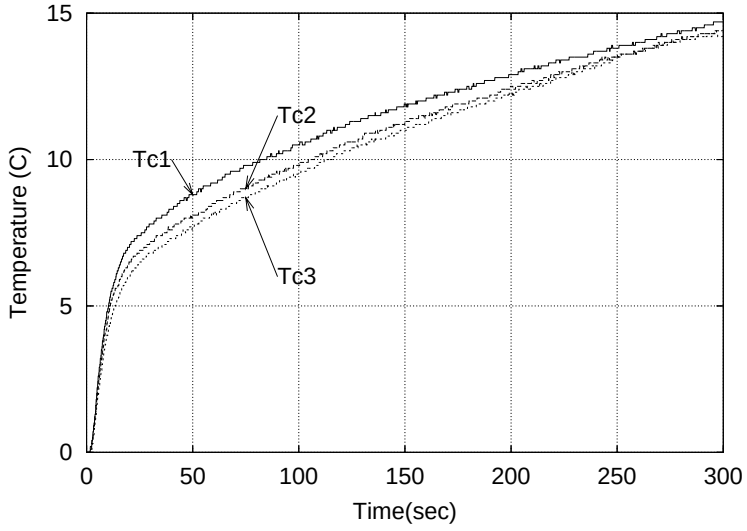
Figure 1.2 (b) shows a open-loop temperature response to a step input in lamp power change. Time constants for quartz windows are in the order of few hundred seconds while the time constants for the wafer typically ranges from below 5 seconds at above 700°C to around 15 seconds and larger at around 400°C or lower operating temperature level. While one can include the quartz window heating for wafer temperature dynamics identification, the results of conventional identification are largely dependent upon the data collection time and lamp heating history (lamp power trajectory). As mentioned in previous paragraph, detrending of drift amount is necessary for the open-loop identification and identification results are dependent on the detrending and also on the selection of identification period.

Alternatively, a closed-loop(PI control) identification scheme for modeling the relationship between lamp heating zone power and wafer temperature zones (edge, middle and center) may be preferred to an open-loop identification method due to following reasons.

- Wafer temperature is kept under control
- Data collection time (identification time) can be reduced
- Detrending step of drift data from response data is not needed.
- Required lamp power levels to obtain the same temperature changes in each set points can be automatically applied. In an open-loop step test, one may apply a step change to lamp power without knowing a priori what temperature level response one would get for the output temperature.



(a)



(b)

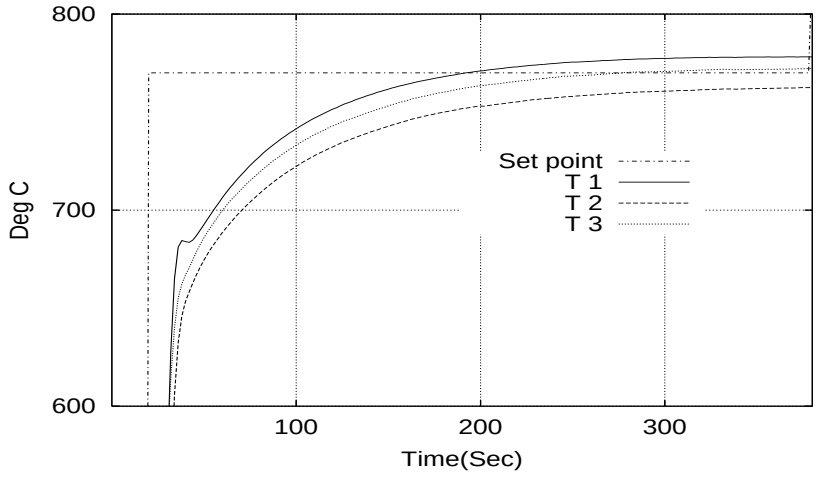
Figure 1.2: (a) Temperature drift at 500 °C , (b) Open-loop Step Responses of 3 Temperatures to 1.42 % change in Lamp Zone 1 Power Level at 700°C

1.3 Ill-conditioned Behavior of RTP

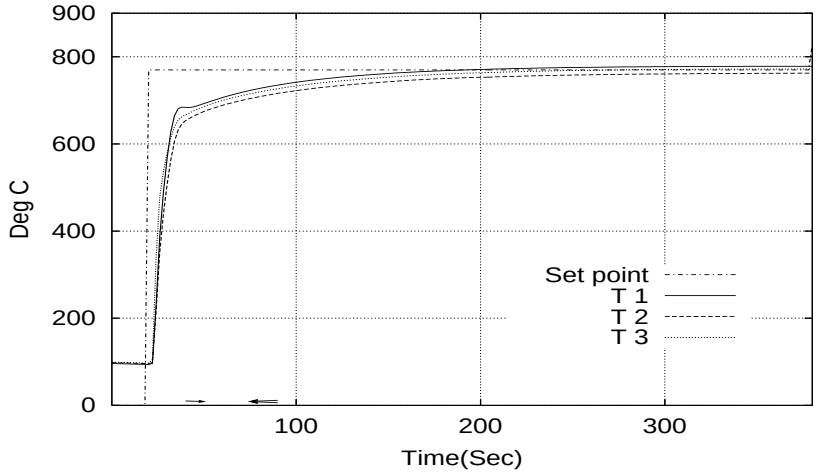
In RTP, the ramp rate of the set points should be as fast as possible and the temperature distribution across the wafer should be uniform(within ±1 °C for processes dealing with critical junction depth). A temperature response plot for a step change in set point from 100°C to 770°C is shown in Figure 1.3. The initial ramp rate was 30°C per second and the temperature difference was about 16 °C across wafer during the hold. The temperature difference (nonuniformity) of the response seems to be largely due to the current lamp housing design and also is related to the ill-conditioning of the multivariable control system. The slow convergence to the set point can also be related to multiloop or multivariable controller tuning. To reduce interactions of lamp zones, several lamp baffle rings can be employed. A baffle ring can be chosen that yields the best temperature uniformity (within 1 °C among 3 temperature measurement points). The condition number of the gain matrices for a typical RTP system without the baffle ring can range from 10^2 to 10^3, which is considered to be an ill-conditioned system. Accurate identification of the process for ill-conditioned systems would be very important especially for multivariable controllers based on the inverse of the model gain matrix. However, on-line identification is more difficult because of the ill-conditioned nature of RTP control systems.

1.4 RTP Temperature Measurement

Wafer temperature measurement based on radiation pyrometry [22] is widely used for commercial RTP chambers. Pyrometers with narrow band gap filters can



(a)



(b)

Figure 1.3: (a) Details of wafer temperature responses to step changes in set point, (b) Temperature response curves for step change in set points from 100°C to 770°C

be used to measure wafer temperature and filter out infrared light interferences from radiation power sources(from lamps used for RTP system, such as Tungsten-Halogen lamps). Since emissivity of silicon wafer is dependent on temperature, the emitted signal from the heated wafer is a function of this intrinsic property of the silicon wafer. When external layers such as oxide, nitride or metal films are deposited on the measurement side of the wafer, the emissivity could change. The variation of the emissivity by these extrinsic factors can cause temperature measurement errors.

Suppose an RTP recipe is set for production with an emissivity of 0.65 based on standard calibration using SensArray thermocouple instrumented wafer, and a wafer with unwanted film deposited on the measurement side of the wafer has an actual emissivity of 0.45 (decrease of 0.2 in emissivity). The temperature measurement error from the pyrometer can be estimated. If $L(\mu W/(\mu m cm^2)$ is the radiated signal from wafer to pyrometer, ϵ_w is the emissivity of wafer from which the original calibration was performed, ϵ_u is the actual emissivity of the wafer with unwanted deposit, then Planck's distribution law is

$$L_w = \epsilon_w \frac{C_1}{\lambda^5} \frac{1}{\exp^{C_2/\lambda T} - 1}$$

At 900°C , $L_w = 1.5491\ W\mu m^{-1}cm^{-2}$. Since the feedback controller keeps the pyrometer signal at this value(actual set point value assigned for the pyrometer), this value is assumed constant during the hold. Then the actual wafer temperature, T_u, with emissivity ϵ_u at the measurement side of wafer can be estimated as

$$T_u = \frac{C_2}{\lambda} \frac{1}{\log\left[1 + \epsilon_u/\epsilon_w(\exp(C_2/(\lambda T_w) - 1)\right]}$$

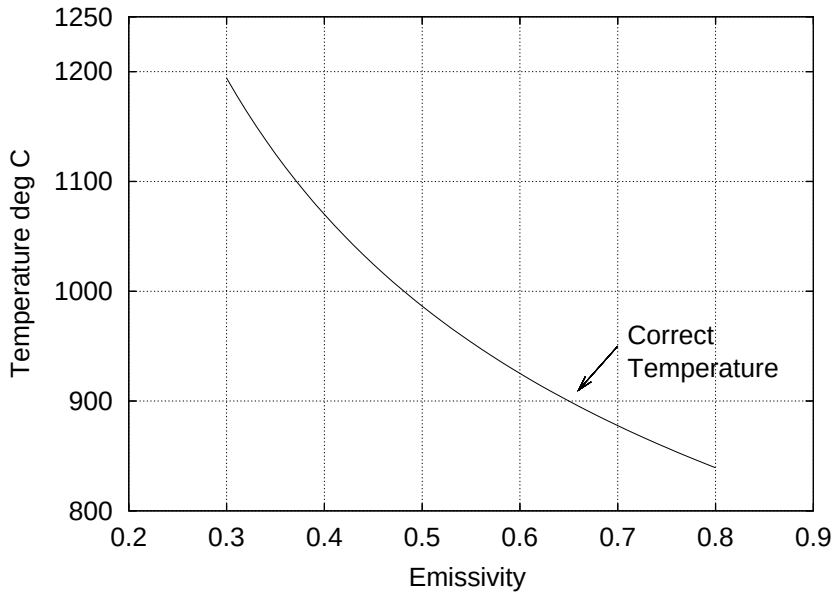


Figure 1.4: Actual wafer temperature for wafer with emissivity ϵ_u when the RTP recipe is set with calibration performed on calibration wafer with emissivity 0.65 at 900°C

Thus for above case, emissivity decrease of 0.2 ($= \epsilon_w - \epsilon_u$) would give $T_u =$ 1024.66°C , with increase in measured temperature, ΔT of 124.66°C . If the recipe set point is 900°C and the wafer with unwanted deposition was processed above 1000°C , this amount of error in temperature measurement is not tolerable for current RTP steps requiring within ±1°C temperature uniformity at hold. Normally, operators or engineers would be able to manually screen out such wafers and properly adjust the recipe and the wafer will go through right process. But in order to avoid misprocessing originating from such pyrometer reading error due to emissivity change, there should be a systematic way of alarming the tool when it happens. Figure 1.4 shows that the temperature error becomes more serious as the emissivity

difference gets larger.

Useful techniques for estimating effective emissivity and related variables could diagnose variation of emissivity due to extrinsic materials deposited on the measurement side or partial area of wafer.

1.5 Controller Design

Multiloop PI controllers can be used for temperature control in RTP. Design of single input single output (SISO) control systems can be easily achieved using techniques such as IMC (Internal Model Control) developed by Morari and coworkers [68], [61]. While multiloop PI controller tuning parameters can initially be set the same as that of individual SISO PI controllers, actual multiloop responses may not be satisfactory due to interactions between input/output variables. A detuning step of individual PI controllers can be performed to avoid unwanted interactions between control loops and to achieve desired responses.

In RTP applications modern multivariable control schemes have been introduced and applied to commercial RTP systems to control wafer temperature; however, many RTP systems such as AG Heatpulse 8108 still use SISO control system. Typically the lamp zone (bank) powers are set up as a percentages of the full power for several lamp heating stages and stored as a recipe parameter values. During temperature control, SISO PI control can be applied to use pyrometer-measured temperature at the center of the wafer as controlled variable and one single manipulated signal is used to control the power of the 9 zones (banks) of lamps. Improved multivariable or multiloop control schemes for RTP are desired to reduce recipe set

up time or to optimize existing recipes. Because enhancement of control performance by updating the recipe is time-consuming, control engineers usually are not able to upgrade the RTP temperature control system. Iterative learning control with multiloop PID control feedback has been applied to RTP control recently, but still this scheme requires the control engineer to obtain signals from each individual control loop. Hence better methods are needed.

1.6 Previous Work on RTP

In 1993, Texas Instrument demonstrated single wafer integrated manufacturing (based on 150 mm wafer) under the MMST (Microelectronics Manufacturing Science and Technology) program sponsored by the Department of Defense (Moslehi [62], Davis [20]). One of the key contributions of the MMST program was to introduce new concepts to semiconductor industry including real time control. Among SWP(Single Wafer Processing) tools, RTP tool is one of the oldest tools introduced (1980s, Singh and Nulman[83]). In 2000, Edgar et al.[24] published a comprehensive review article on applications of advanced process modeling and control in wafer processing including RTP tools. Recently numerous research articles for single-wafer manufacturing technology have been published including application of RTP for manufacturing of advanced flash memory (Chen et al.[15]), batch(furnace) and SWP(RTP) process comparison in memory production by Weimer et al.[104] and feasibility study of RTP as an alternative to batch furnace for 0.15 μm 128-Mb mask read only memory(MROM) production (Hsu et al.[40]). Below present research efforts for RTP technology are briefly introduced with emphasis on modeling, control

and temperature measurement issues which are directly related to this research.

1.6.1 RTP modeling

Lord[55] published thermally induced stress through modeling of heat transport phenomena and also showed that the convective heat transfer coefficient h can be expressed as

$$h\left(\frac{r}{R}\right) = h_i + (h_o - h_i)\left(\frac{r}{R}\right)^4 \tag{1.1}$$

where (r/R) denotes the normalized radius, $h_i = h_{it} + h_{ib}$ (h_i: coefficient at center, t and b denote top and bottom, respectively) and $h_o = h_{ot} + h_{ob}$ (h_o: coefficient at edge). Best fit values for a wafer without shield ring were

$$h_i = 14.2 W/m^2K \tag{1.2}$$

$$h_o = 22.8 W/m^2K \tag{1.3}$$

Lord[55] also showed that with annular shielding ring, the uniformity was improved. Campbell[11] studied gas flow pattern and wafer temperature distribution based on momentum and heat balance equations.

In 1992, Schaper et al.[74] developed a control-oriented low order model relating lamp power zone to wafer temperature zone (3x3 pairing) low order model. The heat balance model with radiation boundary condition and effective emissivity were derived and verified experimentally. Schaper expressed the radiated heat energy to wafer surface as

$$q = -\sigma D T_i^4 + FP \tag{1.4}$$

where effective emissivity D is a function of wafer emissivity(ϵ) and view factor (V: details explained in Chapter 3) due to reflections between wafer zones. Different from other studies which use wafer emissivity ϵ, equation 1.4 employs effective emissivity D and this value can be different from reported wafer emissivity values(Sato[72]) which are measured in black body environment. The value of D also reflects the quartz window heating effect as well as above mentioned radiation reflections between wafer zones.

Park et al.[65] introduced a low order model using the Kalman-Loéve-Galerkin procedure. While the low order model's dimension corresponds to the number of sensor-lamp power pairings (three for the system studied by Schaper et al.[74]), Park et al.[65] expressed the temperature distribution function as a sum of empirical eigenfunctions. They found the number of eigenfunctions that does not increase the error between the finite difference model and the solution based on the sum of the empirical eigenfunctions. An optimal controller based on this finite number of eigenfunctions and boundary conditions was designed in their work.

The purpose of obtaining a good RTP model is to understand various operating characteristics through first principle model and model verification. Most models are intended to design effective lamp configuration(Jan et al.[43]) and advanced feedback controllers for RTP (Lin et al.[53] and papers cited earlier).

1.6.2 RTP Control

Due to the nonlinear (Chatterjee et al.[12]) and ill-conditioned(Breedijk et al.[10], Balakrishnan et al.[5]) nature of the RTP system, the classical SISO PI

controller is not a good choice for RTP temperature control. A gain scheduling multiloop PID controller was implemented by Schaper et al.[75]. Through the SVD analysis, they showed that decentralized controllers can be a good choice for the RTP system, which is ill-conditioned. Different control strategies for RTP systems prior to year 2000 are introduced in a review article by Edgar et al.[24].

Recently, ILC (Iterative Learning Control) scheme introduced by Arimoto et al.[3] has been actively applied to RTP temperature control. Lee et al.[51] added feedback control to ILC as

$$u_{k+1}(t) = u_k(t) + \Gamma(t)(y_d(t) - y_k(t)) + \text{feedback} \tag{1.5}$$

where $u_k(t)$ and $y_k(t)$ are the controller output and output variable at kth iteration step at time t, respectively. $y_d(t)$ is a set point trajectory and $\Gamma(t)$ is a learning gain matrix. Since the controller output is updated based on the previous controller output and current feedback signal, ILC with feedback control scheme results in good control responses to batch processes with repeated cycle.

In RTP temperature control, the overshoot at the ramp and hold turnover can be reduced by ILC scheme. Choi et al.[19] applied ILC scheme to RTP temperature control by utilizing neural networks to estimate the learning gain and also feedforward neuro controller was implemented in their work to estimate the various reference profile from accumulated controller output history. Yang et al.[105] applied quadratic optimal criterion to ILC to control an 8inch wafer RTP system.

For more advanced control approach where the controller adapts its control behavior to varying RTP system parameters, Qin et al.[67] applied adaptive run-to-

run control techniques to industrial RTA (Rapid Thermal Annealing) process and improved RTP tool performance.

1.7 Summary of Research

To address the issues described above for RTP modeling, instrumentation, identification and control, the following topics have been investigated and the results reported in this presented in this dissertation.

- An RTP chamber for experimental testing was set up. This system allows 0.1 Hz fast sampling rate temperature control under 3 x 3 multi-input multi-output control configuration, with temperatures as controlled variables and lamp zone voltages as manipulated variables. Experimental details are described in Chapter 2 and Appendix A.

- In Chapter 3, a rigorous heat transport model is obtained by applying various radiation boundary conditions. This model was discretized for numerical modeling. Using the model, an optimization step was developed to estimate effective wafer emissivity. Using the empirically estimated process parameters, the effective wafer emissivity can be estimated on-line and this information can be utilized to screen out wafers with unwanted backside (measurement side) deposition.

- In Chapter 4, a closed-loop identification method is introduced and actual parameter identification experiments were performed based on this method. This identification scheme is considered to be superior to the open-loop method be-

cause the identification is performed by changing set points, not by changing the controller outputs. In this way, one can avoid trying several lamp power changes to find compatible operating temperatures for each loop. Also, as described previously, model identification time can be reduced. Actual identification experiments were performed with set points ranging from 300°C to 800°C .

- The RTP system without a baffle ring was found to be highly ill-conditioned in nature. Multivariable control schemes utilizing model gain matrix (with matrix inversion) may not function properly if the identified parameters are not accurate. In Chapter 5, an iterative identification method was developed to identify an ill-conditioned system as accurately as possible. In Chapter 6, effects of diagonal input and element uncertainties on identification and control of the ill-conditioned system were discussed in detail.

- Various multiloop PI controller design methods are introduced in Chapter 7, and the selected Multiloop PI controller is used for experimental testing.

- Finally in Chapter 8, an SISO iterative control scheme was applied to RTP temperature control. Different from other multiloop PI controllers or multivariable controllers, this controller uses only one pyrometer signal and one lamp power level for controller output. Other lamp power levels are calculated from this representative measured signal-lamp power pairing. This method is simpler and easier to use for RTP tools with various production recipes where recipe optimization is not an easy task. Therefore this method can system-

atically optimize the recipes for certain industrial RTP systems with SISO control capability only.

- Appendices are included describing hardware and software details for the RTP computer control system as well as some detailed mathematical derivations.

Chapter 2

Experimental Setup

2.1 Overall Description

The overall functional block diagram of the RT/RPCVD reactor system including PC-PLC supervisory control system is shown in Figure 2.1. The schematic of the experimental setup is shown in Figure 2.2 and actual pictures are shown in Figures 2.3 and 2.4. Wiring details for the data acquisition and control system for the PLC-RT/RPCVD system are shown in Figure A.5. All experimental components shown in Figure 2.2 are located inside the fume hood; if there is a leak of any hazardous gases, the air flow rate into the hood is fast enough to dilute any hazardous gases accidentally.

As shown in Figure 2.2, the RT/RPCVD reactor can be used to deposit PECVD dielectric films using inductively generated plasma; rapid thermal processing can also be carried out in the same chamber. Silane gas is supplied to the gas distribution network through the silane gas purge panel to prevent the silane gas from reacting with air. Other gases(UHP grade N_2 , Ar and NH_3) are supplied directly to the gas distribution network from gas cylinders located in the utility aisle to the gas distribution network through long(15 feet) gas delivery lines passing above the reactor system.

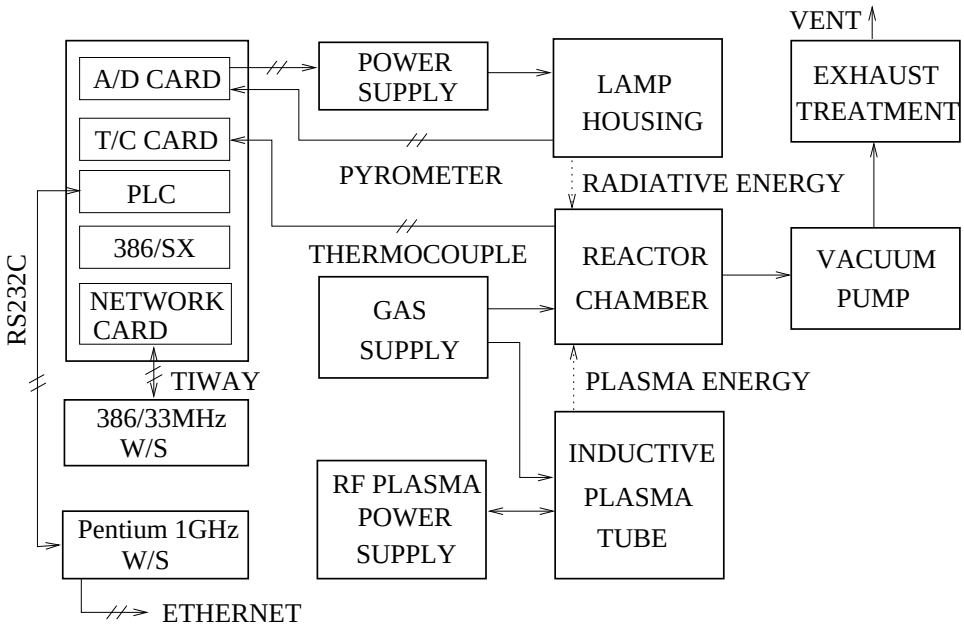


Figure 2.1: Overall block diagram of the RT/RPCVD reactor system

The gases(non-plasma gases, SiH_4 and NH_3 diluted in Ar gas) to be supplied downstream of the inductive plasma region are mixed together at the gas distribution network and transported to the reactor chamber(refer to point Ⓐ in Figure 2.2). SiH_4 gas can be diluted with Ar or N_2 gases at the gas distribution network. Plasma gases(either Ar or N_2) are supplied separately to point Ⓑ of the reactor chamber. Reaction occurs downstream of the inductive plasma and the dielectric films can be deposited at the bottom side of wafer surface.

Inert, unreacted and undeposited gas species flow out of the reactor chamber and pass through the throttle valve, which regulates the pressure of the chamber using

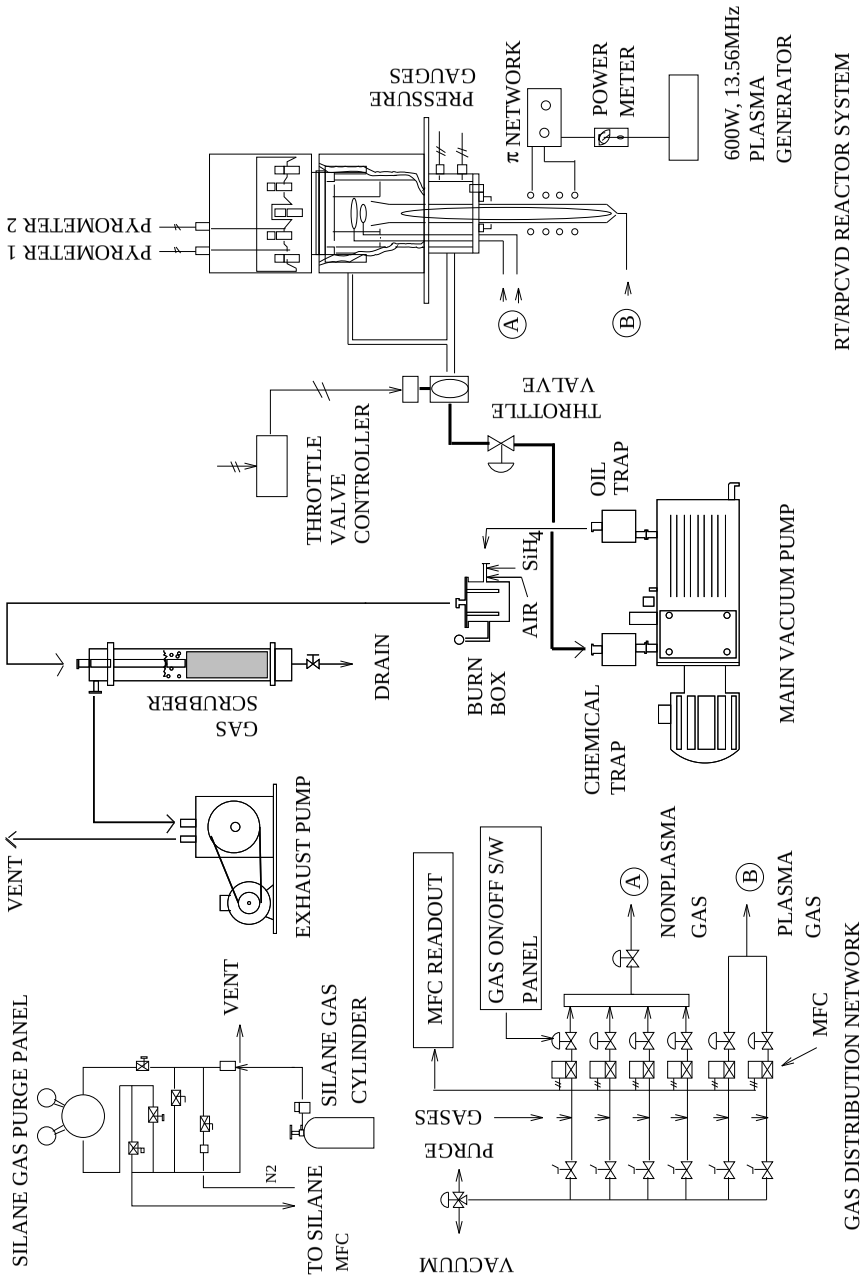


Figure 2.2: Schematic of the RT/RPCVD reactor

pressure measurements from two capacitive manometers located at the bottom side of the chamber. The chemical trap prevents oil(vacuum pump lubricant) back-streaming from the vacuum pump. Gas is pumped out by the vacuum pump and flows through the oil trap, which removes most of oil vapor contained in exhaust gas flowing out of the pump.

Unreacted silane gas in the exhaust gas from the oil trap can be combusted inside the burn box, which should be always kept below atmospheric pressure. Air with a restricted flow rate flows into the burn box through a filter to react with silane gas. So it is important to keep the burn box pressure below atmospheric by applying vacuum from exhaust vacuum pump.

Exhaust gas from the burn box enters a scrubber which is packed with Raschig rings, and filled with 15 weight % sodium hydroxide solution. It is expected that most of the SiO_2 particles are scrubbed here. Finally the exhaust gas from the scrubber enters the exhaust pump and is vented to the fume hood outlet.

Deposition can occur at various temperature ranges. PECVD films can be deposited at temperature ranges from 200 to 400 °C . Rapid thermal processing can be done sequentially at temperature ranges over 900 °C . Temperature uniformity across wafer and transient response control of temperature are important research issues in both steps. Details of radiative heating mechanism, identification and control issues are explained in later chapters.

The inductively coupled plasma(ICP) generation system consists of a *rf* power gener-

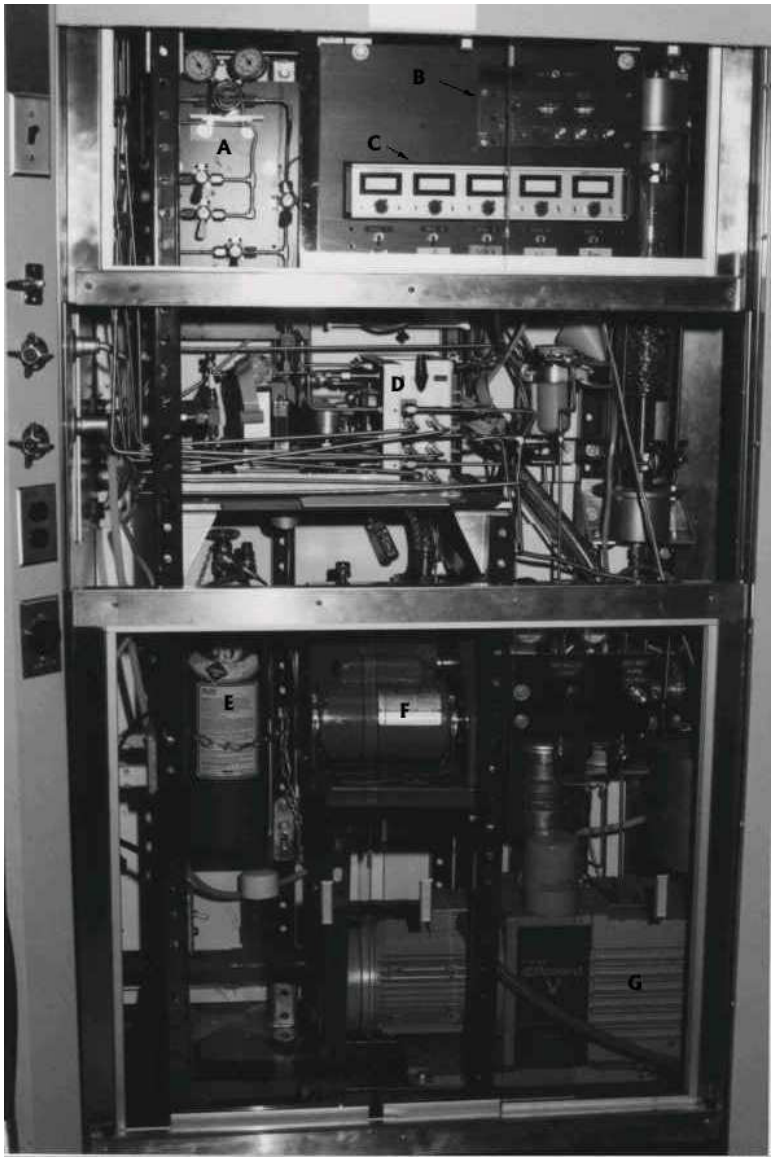


Figure 2.3: Left side of the reactor system

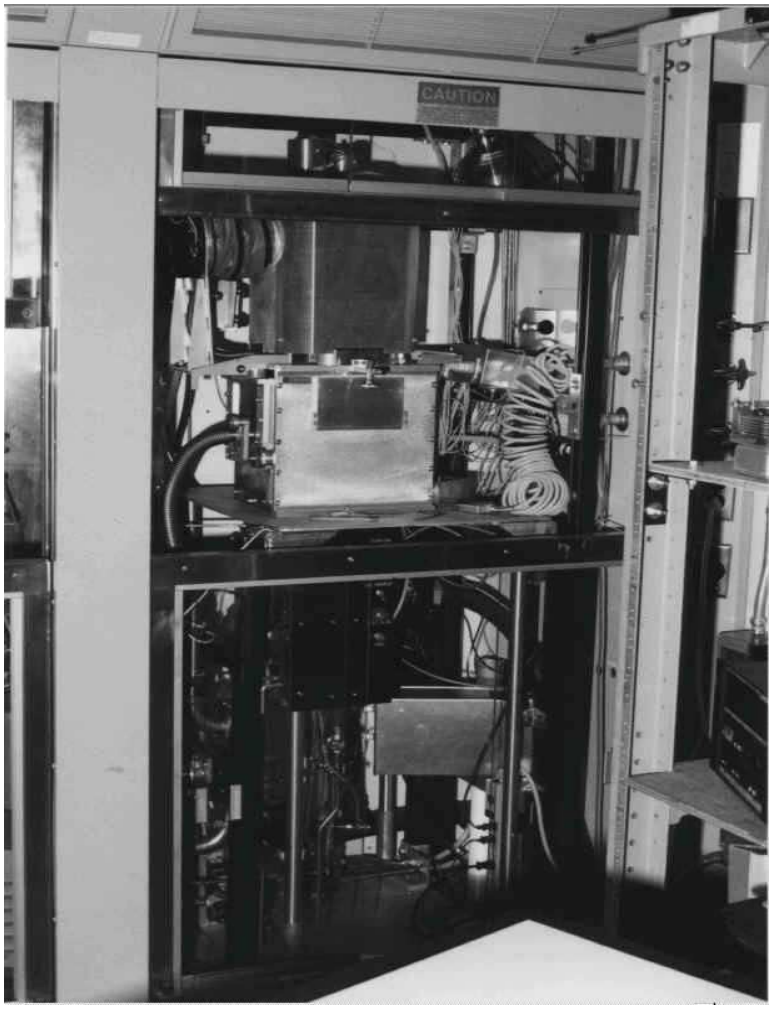


Figure 2.4: Right side of the reactor system

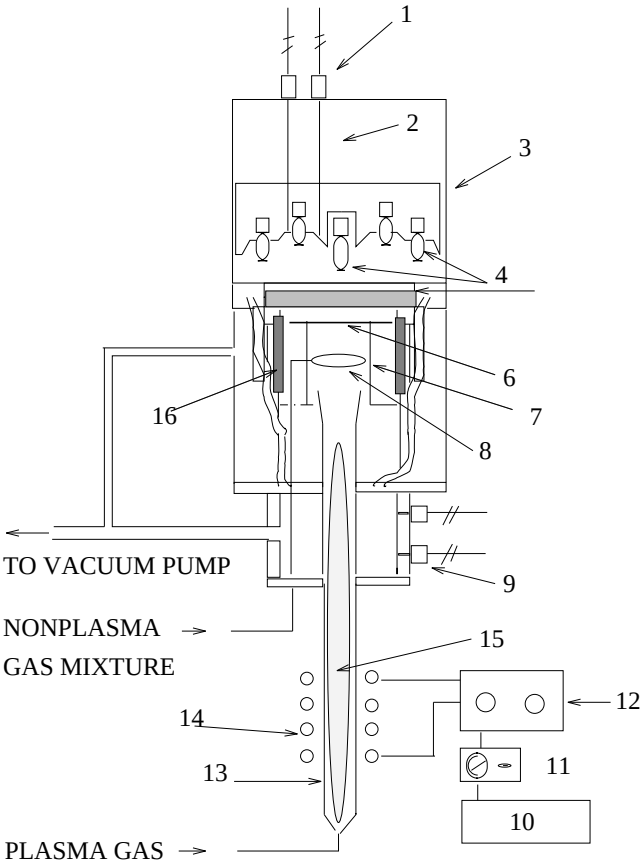
ator, a π circuit impedance matching network, a power meter and a copper inductive coil (Figure 2.5). Details of the ICP system are provided in a later section.

A supervisory control and data acquisition(SCADA) system was built to control the RT/RPCVD process(refer to Figure A.5). A Texas Instruments(TI)[1]'s series 545 PLC is used as a front end controller. A Pentium 1 GHZ(with 512MB RAM, 17GB hard drive) personal workstation which runs on Windows XP operating system is connected to 545 PLC through a RS232C serial protocol with 19200 bits per second communication speed.

Process variable values, such as thermocouple and pyrometer readings, are read by the proper interface cards in the PLC rack and transferred to the PLC through a VME(Versa Modulo Europa) bus located at the backplane of the PLC rack. The PLC performs necessary calculations including PID loops and sends out signals to actuators, such as lamp power supplies, through the D/A(Analog Out) converter card.

In the personal workstation, a program written in Borland C(version 4.0) under Microsoft Windows environment carries out control and identification tasks by sending commands to the 545 PLC and receiving(reading) data from the PLC memory locations through a RS232C serial communication. Process data read by PLC and transferred to the workstation are written to the hard drive in real time at the proper sampling rate. The data written on the hard drive can be retrieved later for analysis.

[1]The programmable logic controller(PLC) division of TI has been merged to Siemens in early 1990's after the PLC was purchased



1. OPTICAL PYROMETER 2. LIGHT PIPE 3. LAMP HOUSING
4. TUNGSTEN/HALOGEN LAMPS 5. QUARTZ WINDOW
6. 6 INCH WAFER 7. QUARTZ PIN 8. GAS RING
9. PRESSURE TRANSDUCER 10. RF POWER GENERATOR
11. POWER METER 12. IMPEDANCE MATCHING BOX
13. PYREX TUBE 14. COPPER INDUCTION COIL
15. PLASMA 16. COLD WALL

Figure 2.5: Details of the reactor system

The RT/RPCVD reactor was purchased from Texas Instruments(TI) in Dallas, Texas. The reactor was built using one of the designs used for Texas Instruments' MMST(Microelectronics Manufacturing Science and Technology) project[62]. The reactor chamber was built and helium leak tested at the M & D Machine & Tool, Inc. at Garland, Texas. The lamp housing including the gold plated reflector and tungsten halogen lamps and the gas distribution network were also built at TI.

The RT/RPCVD reactor system was originally equipped with 2.45 GHz microwave cavity remote plasma generator. The plasma gas flowed through the 1 inch OD quartz tube which was held horizontally and wrapped by a microwave cavity. The tube was bent 90 degrees upward after it passed the micro-cavity. Some deposition experiments were done, but the growth rate was not fast enough. It is possible that the tube diameter and the bend caused the recombination of the active species.

Hence the microwave remote plasma system was removed and a vertical type 2 inch ID inductive plasma system using radio frequency(13.56MHz) was built and attached to the bottom of the reactor. The matching network and coil were built, and adaptation of the Pyrex tubing to the reactor bottom and the gas inlet were done as a part of this research. The reactor bottom flange was modified and glass to metal adapters were welded to the plasma tube. A gas ring for non-plasma gases was also made.

Oil trap, burn box, and gas scrubber were designed by the author and built by Larry Larson in the machine shop at the Chemical Engineering Department. The design for the silane gas purge panel was obtained from David Pierce of the

Electrical and Computer Engineering Department. Orbital welding was used to assemble the elements.

2.2 Reactor System

A detailed layout of the RT/RPCVD chamber was shown in Figure 2.5.

2.2.1 Radiative heating mechanism

Figures 2.6 and 2.7 show a picture and the detailed configurations and actual dimension of the tungsten-halogen lamps, half-inch thick 10 inch diameter quartz plate, and the 6 inch wafer inside the cold-wall reactor. The edge lamp zone(U_1) consists of 24 lamps(1 KW), the middle lamp zone(U_2) consists of 12 lamps(1 KW), and there is one 2 KW lamp in the center(U_3). The reflector is covered with gold plating material to enhance the light reflection. The lamp housing(inside of the reflector) is cooled by water continuously during the experiment. A fan was also installed at the lamp housing cover to enhance convective cooling of the inside of the lamp housing.

Lamp sources and general equipment issues of rapid thermal processing are introduced in detail by Roozeboom and Parekh[71], and Roozeboom[69][70]. Tungsten(W)-halogen lamps are widely used as the radiant energy sources in commercial rapid thermal processors. The lamp consists of a quartz tube(envelope) around a tungsten filament, and the quartz envelope is filled with halogen gas. Due to the tungsten-halide regenerative transport cycle, at quartz temperatures above 250°C the halogen prevents the deposition on the envelope which suppresses the

envelope blackening. The presence of halide gas increases the lifetime of the lamps. Other heat sources such as *noble gas long-arc lamp* and *resistively heated bell jar*, are used commercially.

According to Wien's displacement law

$$\lambda_{\text{peak}} T = 2.89783 \times 10^{-3} \qquad m \cdot K. \tag{2.1}$$

From equation 2.1, we can estimate the black body temperature of the tungsten filament when $\lambda_{\text{peak}} = 1.0\ \mu m$, as 2900 K. The quartz envelope heats up to 400°C with maximum allowable temperature being 900°C . Warm quartz can be a secondary heat radiation source with $\lambda_{\text{peak}} \geq 5 \mu m$.

2.2.2 Temperature measurements

In this research, similar to other temperature control experiments, wafer temperatures at different wafer points were measured using both a thermocouple instrumented wafer and infrared pyrometers.

SensArray's series 1501 thermocouple instrumented wafer(6 inch) is used to measure the temperatures at various points on the wafer surface[79]. Chromel-Alumel (type K, temperature range 0 °C to 1100 °C) thermocouple wires are bonded with high temperature ceramic cement(composed of 60 % SiO_2 and 40 % Al_2O_3) into the wafer surface. The various locations of the thermocouple junctions on the wafer and actual picture are shown in Figure 2.7 and 2.8.

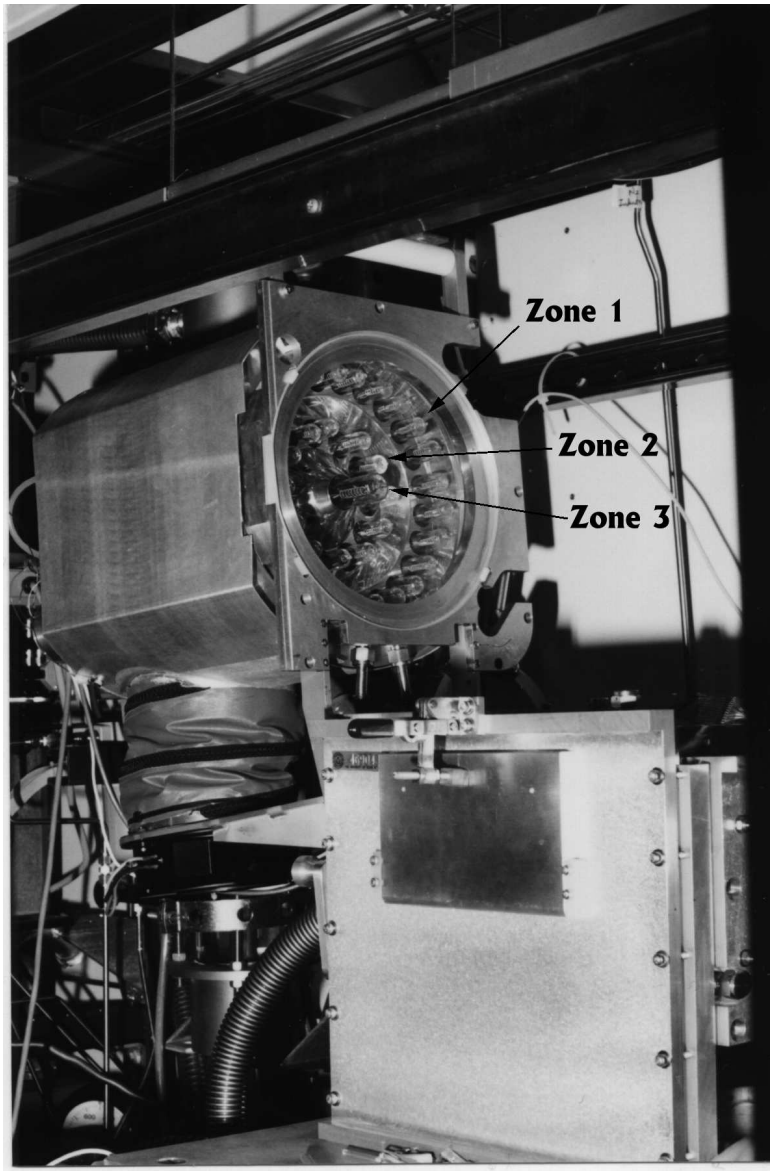


Figure 2.6: Lamp housing flipped open to the left

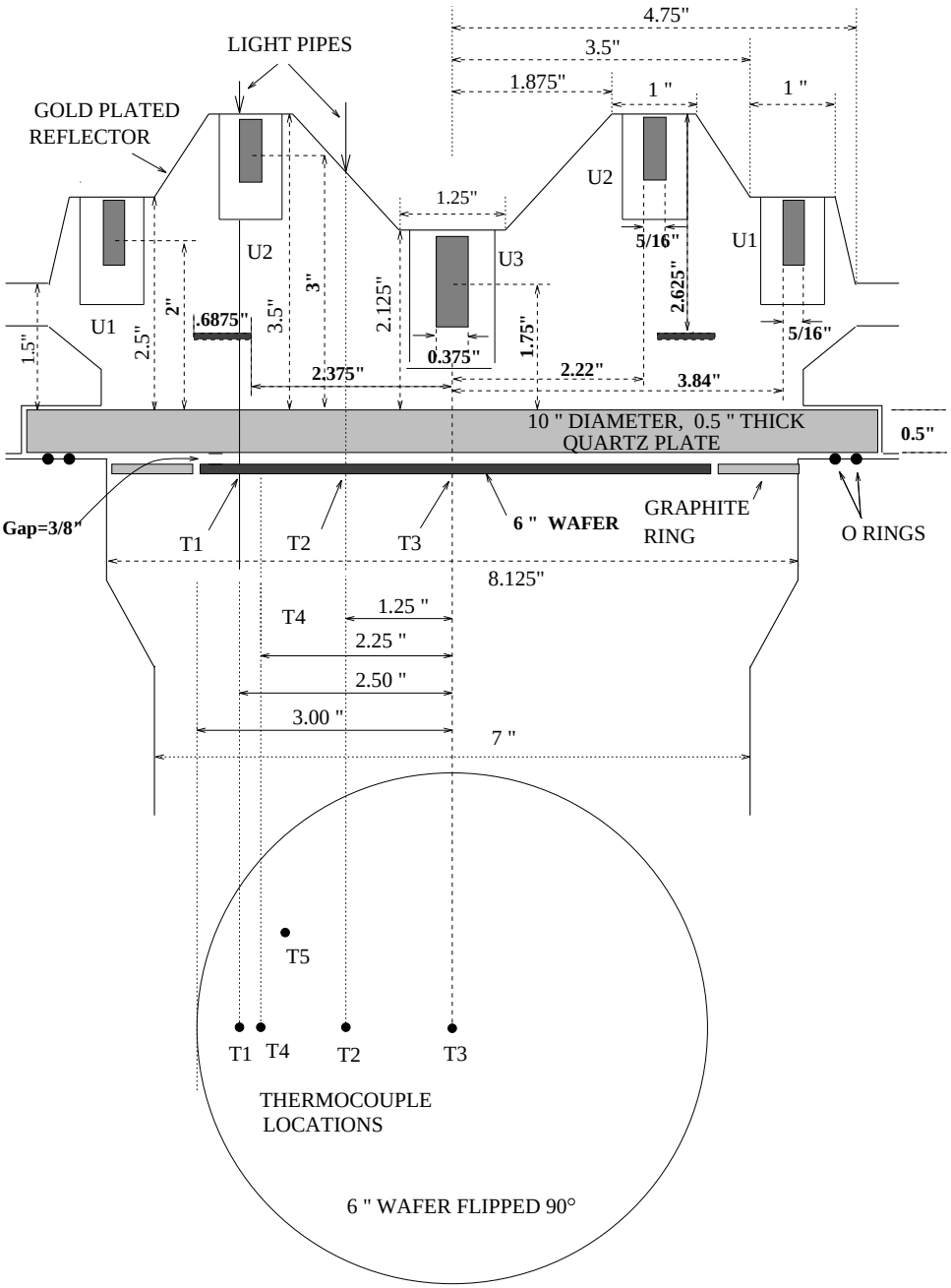


Figure 2.7: Details of the reactor; lamp quartz plate, and wafer

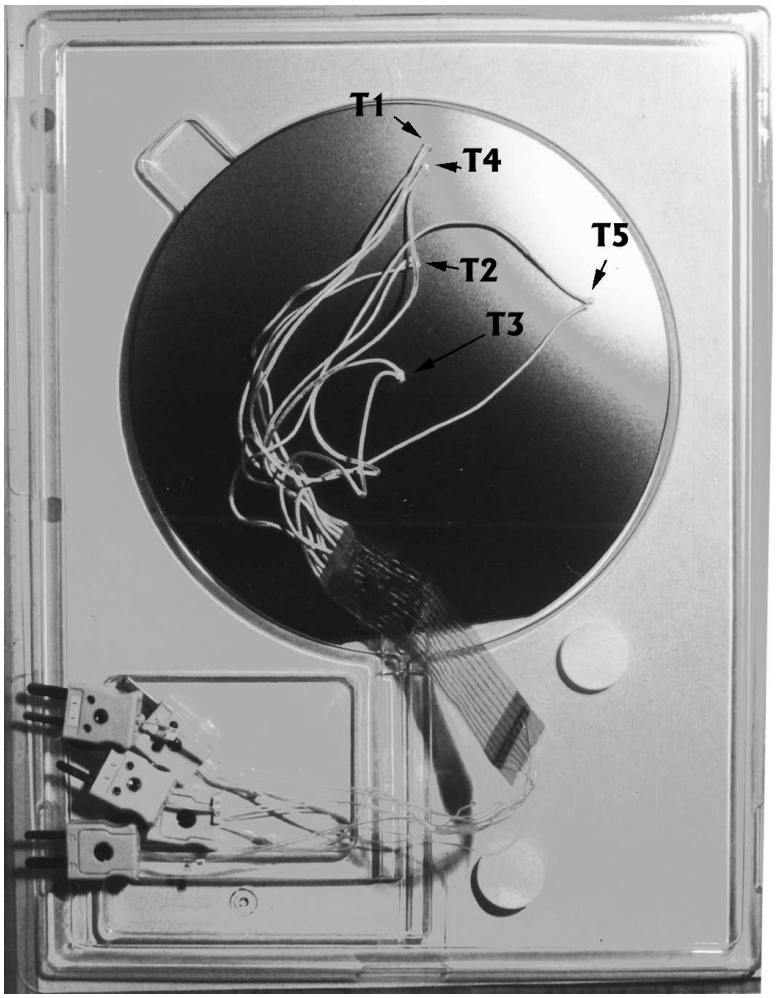


Figure 2.8: SensArray thermocouple instrumented wafer

2.2.3 Inductively coupled plasma generation

As explained previously, Ar plasma is generated by the ICP generation system shown in Figure 2.9. The *rf* generator installed is model RFX 600 of Advanced Energy Industries, Inc.(Fort Collins, Colorado). Bird Electronic Corp.(Cleveland, Ohio)'s 4411 Thruline Wattmeter with 4410-3 detector element is used to measure forward and reflected powers. Air variable capacitors, C1(506 pF) and C2(800 pF) were purchased from Oren Elliot Products(Edgerton, Ohio). A 1/8 inch diameter copper coil was purchased from a hardware shop and 2 inch ID Pyrex tubing was obtained from the Chemistry Stockroom. Figure 2.10 shows a picture of the plasma tube. The RFX 600 power generator[1] is a two-stage power generator using a Fetpower^{TM} modular power amplifier and a switch-mode dc power supply for main power and control. It is designed to handle 80% reflected power. It regulates output power using either forward power or dc bias of the load. It can be used alone or in combination with other supplies (through common exciter mode) in such configurations as *rf* with *rf* bias, *rf* with dc bias, and dc with *rf* bias. The power level(0 to 600 watts) and pulse mode can be remotely controlled interfaced to PLC cards. It is air cooled.

Plasma enhanced chemical vapor deposition of silicon nitride film was initially attempted. The deposition rate and uniformity were not satisfactory, so further plasma experiments were not performed in this research.

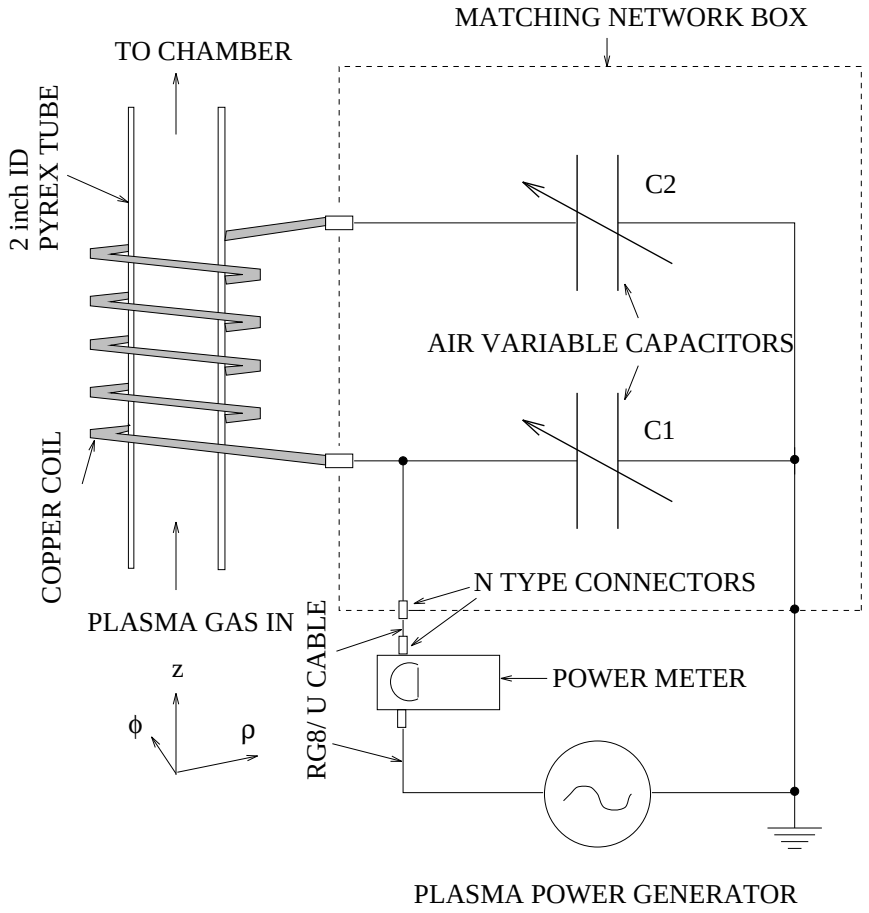


Figure 2.9: Schematic of inductively couple plasma generation system

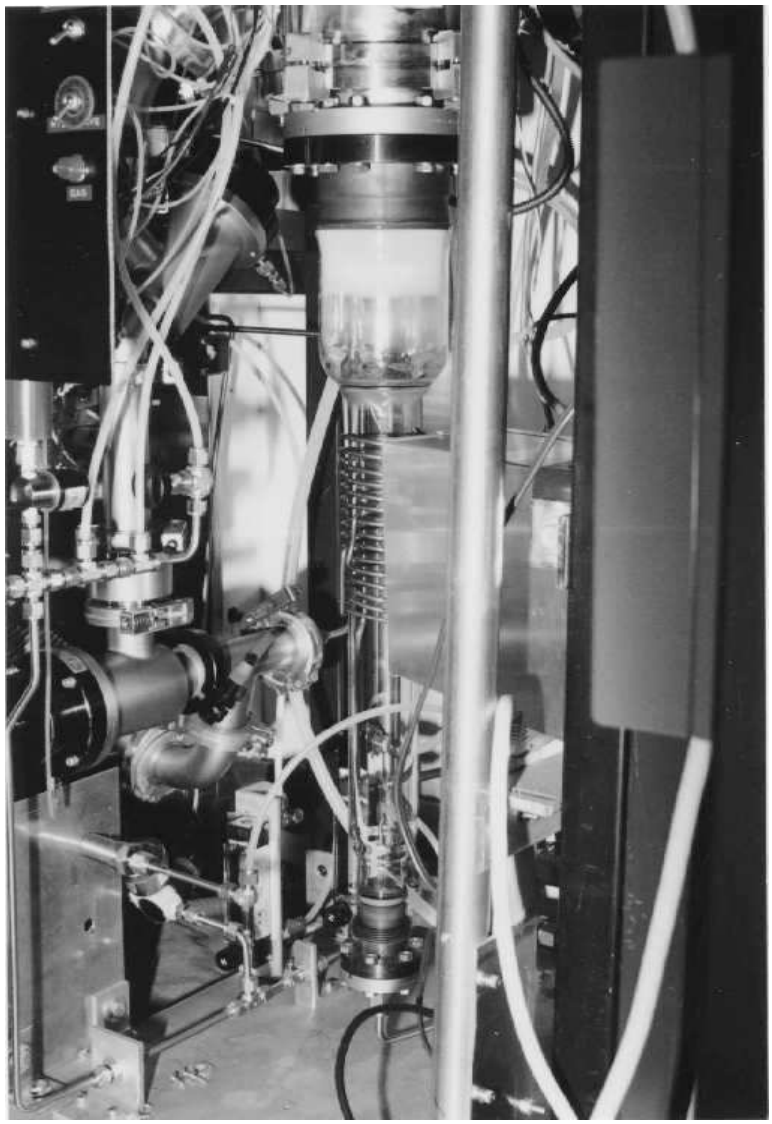


Figure 2.10: Inductive plasma tube

Chapter 3

Modeling of the Wafer Heat Transport Phenomena

Modeling of heat transport phenomena of RTP is needed to characterize the RTP chamber. One of the key design factors of RTP chamber is the configuration of the lamps. For example, current RTP chamber lamp housing consists of lamp reflector plate and 37 tungsten halogen lamps positioned in 3 concentric zones(as described in detail in Chapter:2). Depending on the physical sizes and locations of each lamps and design of the reflector plate, and location of wafer, the view factor for the radiation can change. View factor directly influences process gains which affects wafer temperature response to lamp power change. In some RTP systems, lamp housing design does not allow for the wafer temperature profile to be uniform.

Our modeling work includes view factor between lamp zones and wafer zones, view factor between two wafer zones (through various reflections from lamp reflector plate and quartz window) in the modeling equations. Experimental estimation of these view factors will give engineers information about current lamp configuration and one can use the knowledge to improve the future RTP chamber design.

Accurate model can also be utilized for model based control. Whether it is a simple model based PI control or advanced model predictive control, the model can generate required data for control task.

As was described in Chapter 2, the 6 inch wafer is heated by the radiation energy flux delivered to the surface of the wafer. A heat energy balance equation will be formulated based on Fourier's law of heat conduction, radiation from the tungsten-halogen lamps, emission, reflection and transmission from the wafer surface, reflection by lamp reflector of energy flux from lamps and wafer surfaces and convection occurring on top, bottom and edge side of the wafer. Lamp zones 1 and 2 are modeled to be uniform ring shaped heat sources and lamp zone 3 is assumed to be a circular disk heat source to simplify the modeling process. View factors $(F(r, d_j, l_j))$ from a differential element of the wafer to a ring shaped lamp zone or a circular disk shape are calculated as a function of wafer radius r, inner radius of ring d_j, and ring width l_j.

3.1 Basic Equation Formulation

Modeling of the RTP wafer energy balance has been done before by various authors [10], [74], [76], [65], [41]. Following the energy balance described in standard reference [7], the heat balance equation will be formulated based on following energy transport elements in Figure 3.1 with axisymmetric assumption.

1. Rate of thermal energy accumulation in the element:

$$\rho C_p r \Delta\theta \Delta r \Delta Z \frac{\partial T}{\partial t}$$

2. Rate of thermal energy in across surface with area $r\Delta\theta\Delta z$:

$$(r\Delta\theta\Delta z)(q_c|_r)$$

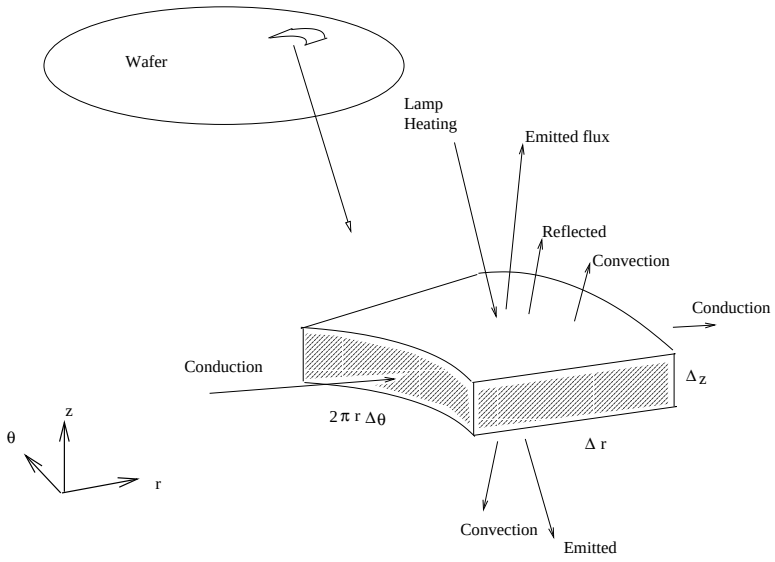


Figure 3.1: Schematic of a wafer differential element and various heat fluxes

3. Rate of thermal energy out across surface with area $r + \Delta r \Delta\theta \Delta z$

$$((r + \Delta r)\Delta\theta\Delta z)(q_c|_{r+\Delta r})$$

4. Rate of thermal energy in across surface at $z = z + \Delta z$:

$$(r\Delta\theta\Delta r)(q_c|_z)$$

5. Rate of thermal energy out across surface at $z = z$:

$$(r\Delta\theta\Delta r)(q_c|_{z+\Delta z})$$

Boundary conditions are:

1. Radiation energy from lamp zones to the top surface of the element

$$q_{r-l}|_{r,Z} = \sum_j F_j(r) P_j \tag{3.1}$$

where matrix F includes direct view factors from lamps(lamp zones) to the annular zones on the wafer as well as the view factors for reflections between two annular zones of the wafer [74].

2. Radiation energy loss from top ($z = Z$) and bottom ($z = 0$) of the wafer surface, assuming $T|_{z=Z} \simeq T|_{z=0}$ for $0 < r < R$,

$$q_{rw}|_{z=Z} = \sigma D(\epsilon) T(r)^4 \tag{3.2}$$

$$q_{rw}|_{z=0} = \epsilon(T) \sigma T(r)^4 \tag{3.3}$$

where matrix D in equation 3.2 is explained in detail in [74]. Matrix D includes view factors of reflections between two annular zones on the wafer. Off diagonal terms of D are small and ignored in the paper([74]).

3. Energy loss due to convection from the top of the element to the ambient gas

$$k \frac{\partial T}{\partial z}\bigg|_{conv,z=Z} = -h_t (T - T_a) \tag{3.4}$$

4. Energy loss due to convection from the bottom of the element to the ambient gas

$$k \frac{\partial T}{\partial z}\bigg|_{conv,z=0} = h_b (T - T_a) \tag{3.5}$$

5. At $r = 0$,

$$\frac{\partial T}{\partial r} = 0 \tag{3.6}$$

6. At $r = R$,

$$k\frac{\partial T}{\partial r} = -h_e(T - T_a) \tag{3.7}$$

The energy balance thus becomes

$$\begin{aligned}\rho C_p(r\Delta r\Delta\theta\Delta z)\frac{\partial T}{\partial t} &= (r\Delta\theta\Delta z)(q_c|_r) - ((r+\Delta r)\Delta\theta\Delta z)(q_c|_{r+\Delta r}) \\ &\quad +(r\Delta\theta\Delta r)(q_c|_z) - (r\Delta\theta\Delta r)(q_c|_{z+\Delta z})\end{aligned} \tag{3.8}$$

Dividing both sides by $\Delta r\Delta\theta\Delta z$ and taking the limit as $\Delta r \to 0$ and $\Delta z \to 0$ gives,

$$\rho C_p r\frac{\partial T}{\partial t} = -\frac{\partial(rq_c)}{\partial r} - r\frac{\partial q_c}{\partial z} \tag{3.9}$$

Using Fourier's law with isothermal conduction property,

$$q_c = -k\frac{\partial T}{\partial r} \qquad \text{and} \qquad q_c = -k\frac{\partial T}{\partial z}$$

equation 3.9 becomes,

$$\rho C_p\frac{\partial T}{\partial t} = k\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r}) + \frac{\partial^2 T}{\partial z^2}\right) \tag{3.10}$$

Equation 3.10 can be integrated across wafer height Z from $z = 0$ to $z = Z$ to give,

$$\left[\rho C_p\frac{\partial T}{\partial t}\right]_0^Z = \left[k\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r})\right)\right]_0^Z + \left[k(\frac{\partial T}{\partial z})\right]_0^Z \tag{3.11}$$

Thus

$$\rho C_p\frac{\partial T}{\partial t} = k\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r})\right) + \frac{k}{Z}\left.\frac{\partial T}{\partial z}\right|_0^Z \tag{3.12}$$

3.1.1 Boundary conditions

Suppose that the wafer is at temperature T(Kelvin). At the wafer top surface, $z = Z$, and radial location(or wafer annular zone) r_i (as in Figure 3.2) , the radiated heat transport boundary condition can be expressed as in [74]

$$k\frac{\partial T}{\partial z}\bigg|_{z=Z} = q_{r_i-r_j}|_{z=Z} + q_{r_i-P_k}|_{z=Z} - q_{r_i}|_{z=Z} + k\frac{\partial T}{\partial z}\bigg|_{conv,z=Z} \quad (3.13)$$

$$k\frac{\partial T}{\partial z}\bigg|_{z=0} = q_{r_i}|_{z=0} + k\frac{\partial T}{\partial z}\bigg|_{conv,z=0} \quad (3.14)$$

where, the 1st term in equation 3.13 is the heat radiated from wafer zone j to wafer zone i through reflection from the gold plated reflector, 2nd term is the heat flux from lamp zone k to wafer zone i, the 3rd term is heat radiated by emission from top of the wafer $z = Z$, and the last term of equation 3.13 is the heat lost through convective transfer at the top of the wafer to ambient temperature. The 1st term in equation 3.14 is the heat lost by emission from bottom side of wafer, and last term is the heat lost from bottom of the wafer through convective heat transfer.

Representing the reflection coefficient from wafer zone j to i as V_{ij}, heat flux from zone j as R_j, view factor from lamp zone k to wafer zone i as W_{ik}, heat flux from kth lamp zone as P_k, emissivity of the wafer as ϵ, reflectivity as ρ, transmissivity as τ, h_t and h_b as the convective heat transfer coefficients from top and bottom of the wafer, respectively, the boundary condition can be expressed as below

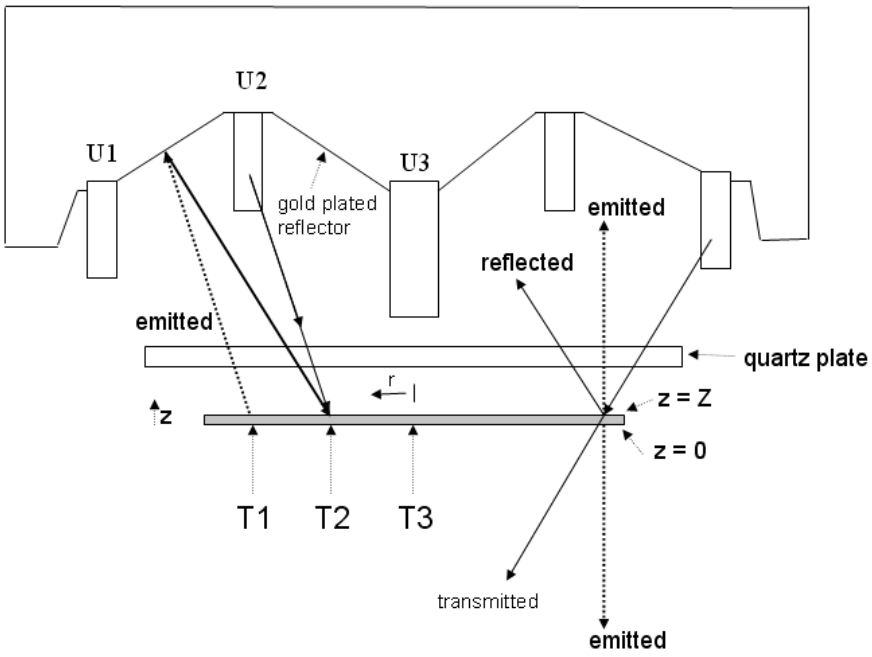


Figure 3.2: Schematic of different types of radiated heat transport mechanism in the RTP chamber

$$q_{r_i - r_j}\big|_{z=Z} = \sum_j V_{ij} R_j$$

$$q_{r_i - P_k}\big|_{z=Z} = \sum_j W_{ik} P_k$$

$$q_{r_i}|_{z=Z} = \epsilon\sigma T_i^4 + \rho(T_i)S_i$$

$$q_{r_i}|_{z=0} = \epsilon\sigma T_i^4 + \tau(T_i)S_i$$

$$k\frac{\partial T}{\partial z}\bigg|_{conv,z=Z} = -h_t(T_i - T_a)$$

$$k\frac{\partial T}{\partial z}\bigg|_{conv,z=0} = h_b(T_i - T_a)$$

Thus equation 3.12 becomes

$$\begin{aligned} Z\rho C_p\frac{dT_i}{dt} &= kZ\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r})\right) \\ &+ \sum_j V_{ij}R_j + \sum_k W_{ik}P_k \\ &-2\epsilon(T_i)\sigma T_i^4 - [\rho(T_i) + \tau(T_i)]S_i - h_w(T_i - T_a) \end{aligned} \tag{3.15}$$

Let

$$S_i = \sum_j V_{ij}R_j + \sum_k W_{ik}P_k \tag{3.16}$$

$$R_i = 2\sigma A_\epsilon(T_i)T_i^4 + [\rho(T_i) + \tau(T_i)]S_i \tag{3.17}$$

where S_i and R_i are the radiative energy flux impinging on wafer, and exiting from the wafer, respectively. Following the procedure and notation in [74], i.e. $T^n =$

$[T_1^n, \cdots, T_N^n]^T, A_\epsilon = diag[\epsilon(T_1), \cdots, T_N],\ A_{\rho,\tau} = diag[\rho(T_1) + \tau(T_1), \cdots, \rho(T_N) + \tau(T_N)]$, and $V^T = V$(V is symmetric) it can be easily derived that

$$R = 2\sigma(I - A_{\rho,\tau}V)^{-1}A_\epsilon T^4 + (I - A_{\rho,\tau}V)^{-1}A_{\rho,\tau}WP \tag{3.18}$$

$$S = 2\sigma(I - VA_{\rho,\tau})^{-1}VA_\epsilon T^4 + (I - VA_{\rho,\tau})^{-1}WP \tag{3.19}$$

Thus equation 3.15 becomes

$$\begin{aligned} Z\rho C_p\frac{dT_i}{dt} &= kZ\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r})\right) \\ &\quad -\sigma DT_i^4 + FP - h_w(T_i - T_a) \end{aligned} \tag{3.20}$$

where

$$D = 2\Big((I - A_{\rho,\tau}V)^{-1} - (I - VA_{\rho,\tau})^{-1}V\Big)A_\epsilon \tag{3.21}$$

$$F = \Big((I - VA_{\rho,\tau})^{-1} - (I - A_{\rho,\tau}V)^{-1}A_{\rho,\tau}\Big)W \tag{3.22}$$

Since $A_{\rho,\tau}$ and $A_\epsilon = I - A_{\rho,\tau}$ are diagonal matrices and all wafer zone temperatures are considered to be at same temperature T, and also assuming that the off diagonal elements of V matrix is negligible (V is a diagonal matrix), the above equations can be simplified as

$$D = 2(I - A_{\rho,\tau}V)^{-1}(I - V)A_\epsilon \tag{3.23}$$

$$F = (I - A_{\rho,\tau}V)^{-1}A_\epsilon W \tag{3.24}$$

When the radial temperature gradient is further ignored because the temperatures at adjacent wafer zones are so close each other near steady state, the heat conduction term in equation 3.25 can be ignored to give,

$$Z\rho C_p \frac{dT_i}{dt} = -\sigma D T_i^4 + FP - h_w(T_i - T_a) \tag{3.25}$$

3.2 Model Linearization

Assuming that temperature uniformity has been achieved throughout the wafer, the heat conduction term (1st term of right hand side) of equation 3.15 disappears.

$$k\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial T}{\partial r})\right) = 0 \tag{3.26}$$

This assumption allows simplified derivation of linear model equation, but as shown below it will also force off-diagonal dynamic terms to disappear from the original dynamic equation. i.e.,

Also, expressing $T(r_i, t) = T_i(t)$ and $F_j(r_i) = F_{ij}$ (where $r_i = \frac{R}{N}(i-1), N$ =number of equally spaced radial points), equation 3.25 becomes

$$\rho C_p \frac{dT_i(t)}{dt} = \frac{1}{Z}\left[\left(\sum_j F_j(r)P_j(t) - \sigma \sum_k D_{ik}(\epsilon)T_k(t)^4\right) - h_w(T_i(t) - T_a)\right] \tag{3.27}$$

Parameters D_{ik} $k = 1, 2$ and 3 cannot be individually identified. We can assume $T_{i1} \approx T_{i2} \approx T_{i3}$, and estimate $\sum_k D_{ik}(\epsilon)$ as one parameter,

At steady state, expressing $T_i = T_{s,i}$ and $P_j = P_{s,j}$, equation 3.27 becomes

$$\rho C_p \frac{dT_{s,i}(t)}{dt} = \frac{1}{Z}\left[\left(\sum_j F_j(r)P_{s,j} - \sigma D_{i,sum}(\epsilon){T_{s,k}}^4\right) - h_w(T_{s,i} - T_a)\right] \quad (3.28)$$

Let

$$T_i(t) = T_{s,i} + \delta T_i, \quad (3.29)$$

then

$$\begin{aligned} {T_i(t)}^4 &\approx {T_{s,i}}^4 + 4{T_{s,i}}^3\delta T_i \\ {T_i(t)}^4 - {T_{s,i}}^4 &\approx 4{T_{s,i}}^3\delta T_i \end{aligned} \quad (3.30)$$

Let

$$\mathbf{T_i(t)} = T_i(t) - T_{s,i} \quad (3.31)$$

$$\mathbf{P_j(t)} = P_j(t) - P_{s,j} \quad (3.32)$$

Subtracting equation 3.28 from equation 3.27, and using equation 3.30 and deviation variables 3.31 and 3.32 gives,

$$\begin{aligned} \rho C_p \frac{d\mathbf{T_i}}{dt} &= \frac{1}{Z}\left[\sum_j F_{ij}\mathbf{P_j} - 8D_i(\epsilon)\sigma{T_{s,i}}^3\mathbf{T_i} - \mathbf{h_w}\mathbf{T_i}\right] \\ &= -\frac{1}{Z}\left[4(D_{i,sum}(\epsilon)\sigma{T_{s,i}}^3 + h_w\right]\mathbf{T_i} + \frac{1}{Z}\sum F_{ij}\mathbf{P_j} \end{aligned} \quad (3.33)$$

Equation 3.33 can be rearranged to give,

$$\frac{Z\rho C_p}{4\sigma D_i(\epsilon){T_{s,i}}^3 + h_w}\frac{d\mathbf{T_i}}{dt} = -\mathbf{T_i} + \frac{1}{4\sigma D_i(\epsilon){T_{s,i}}^3 + h_w}\sum F_{ij}\mathbf{P_j} \quad (3.34)$$

For 3 input and 3 output system, equation 3.34 can be expressed in matrix notation,

$$\begin{bmatrix} \tau_1 & 0 & 0 \\ 0 & \tau_2 & 0 \\ 0 & 0 & \tau_3 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{T}}_1 \\ \dot{\mathbf{T}}_2 \\ \dot{\mathbf{T}}_3 \end{bmatrix} = - \begin{bmatrix} \mathbf{T_1} \\ \mathbf{T_2} \\ \mathbf{T_3} \end{bmatrix} + \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \mathbf{P_1} \\ \mathbf{P_2} \\ \mathbf{P_3} \end{bmatrix} \tag{3.35}$$

where

$$K_{ij} = \frac{F_{ij}}{4\sigma D_i(\epsilon) {T_{s,i}}^3 + h_w} \tag{3.36}$$

$$\tau_i = \frac{Z\rho C_p}{4\sigma D_i(\epsilon) {T_{s,i}}^3 + h_w} \tag{3.37}$$

3.2.1 Characteristics of the parameters

In the following some useful relationships between $F_{ij}(T)$, K_{ij} and W_{ij} are verified. From equation 3.36,

$$F_{ij}(\epsilon, T) = K_{ij}\Big(4\sigma D_{i,sum}(\epsilon, T) {T_i}^3 + h_{w,i}\Big) \tag{3.38}$$

Thus for same wafer location i and at same temperature T the ratio of $\frac{F_{ij}}{F_{ik}}$ can be expressed as

$$\frac{F_{ij}(\epsilon, T)}{F_{ik}(\epsilon, T)} = \frac{K_{ij}}{K_{ik}} \tag{3.39}$$

When we further assume V matrix as a diagonal matrix, from equation 3.24

$$\frac{F_{ij}}{F_{ik}} = \frac{(I - A_{\rho,\tau,ii}V_{ii})^{-1}A_{\epsilon,ii}}{(I - A_{\rho,\tau,ii}V_{ii})^{-1}A_{\epsilon,ii}} \frac{W_{ij}}{W_{ik}} = \frac{W_{ij}}{W_{ik}} \tag{3.40}$$

Since W_{ij} is a view factor from a light source j to a location(zone) i on the wafer, it is a constant independent of temperature of the wafer, so equations 3.39 and 3.40 show that the ratio of process gains for same location on wafer from different light sources are kept at same value independent of temperature changes.

3.3 Sensitivity of K_{ij} and τ_i to T

Based on the equations derived above, temperature dependencies of process gain and time constant are shown below. Namely, derivative of process gain with respect to temperature can be calculated using following set of 4 equations as derived in Appendix B. Parameters s, p, q, and r in equation 3.44 are estimated from the estimation steps described in section 3.4. Then the derivative values of each equations 3.44,3.43,3.42 and 3.41 can be calculated following this sequence to give the value of $\frac{\partial K_{ij}}{\partial T_i}$.

$$\frac{\partial K_{ij}}{\partial T_i} = \frac{\left(\frac{\partial F_{ij}}{\partial T_i}\right)(4\sigma D_{i,sum}T_i^3 + h_{w,i}) - F_{ij}(12\sigma D_{i,sum}{T_i}^2)}{(4\sigma D_{i,sum}T_i^3 + h_{w,i})^2} \tag{3.41}$$

$$\frac{\partial F}{\partial T} = -(I - A_{\rho,\tau}V)^{-1}\frac{\partial A_\epsilon}{\partial T}\left(A_\epsilon V(I - A_{\rho,\tau}V)^{-1} + I\right)W \tag{3.42}$$

$$\frac{\partial A_\epsilon}{\partial T} = \frac{(1-R)dr\exp(-\alpha d)\Big(1 - R + rR\exp(-\alpha d)\Big)\left(\frac{\partial \alpha}{\partial T}\right)}{(1 - rR\exp(-\alpha d))^2} \tag{3.43}$$

$$\frac{\partial \alpha}{\partial T} = s\lambda^p T^{q-1}\exp(-7000/T)(q + 7000/T) \tag{3.44}$$

Also, temperature derivative of time constant τ_i (equation 3.37), when $C_p = a + bT$ can be expressed as

$$\frac{\partial \tau_i}{\partial T} = \frac{Z\rho b}{4\sigma D_{i,sum} T_i^3 + h_{w,i}} - \frac{(a+bT)(24\sigma D_{i,sum} T_i^2)}{(4\sigma D_{i,sum} T_i^3 + h_{w,i})^2} \tag{3.45}$$

3.4 Parameter Estimation

The important parameters to be estimated are $K_{s,ij}$, $\tau_{s,i}$, ϵ, F_{ij}, and h_w. Among these parameters, process gain and time constants $K_{s,ij}$ and τ_{s_i} would be estimated through series of identification experiments at different temperature levels. Emissivity, ϵ, of silicon wafer used is a function of wafer temperature, doping concentration and the dopant type(N or P). Specific heat capacity C_p is also dependent on temperature (as shown in Table B.1). F_{ij}, ϵ, and h_w should be estimated from the experimentally identified parameters $K_{s,ij}$ and $\tau_{s,i}$ using equations 3.36 and 3.37. Since there are 3 parameters to be estimated and only 2 equations are available, following procedure is used for parameter estimation. Figure 3.3 describes the program flow for parameter estimation schematically.

3.5 View factor calculation

Let the view factors from a differential element located at radial distance r on the wafer to each lamp zone j be expressed as $F_j(r)$ (as shown in Figure 3.4). These view factors are tabulated [81] and easily calculated from other basic view factors, i.e., $F_3(r)$ is listed in the table, but $F_1(r)$ and $F_2(r)$ can be calculated from $F_3(r)$ as shown below.

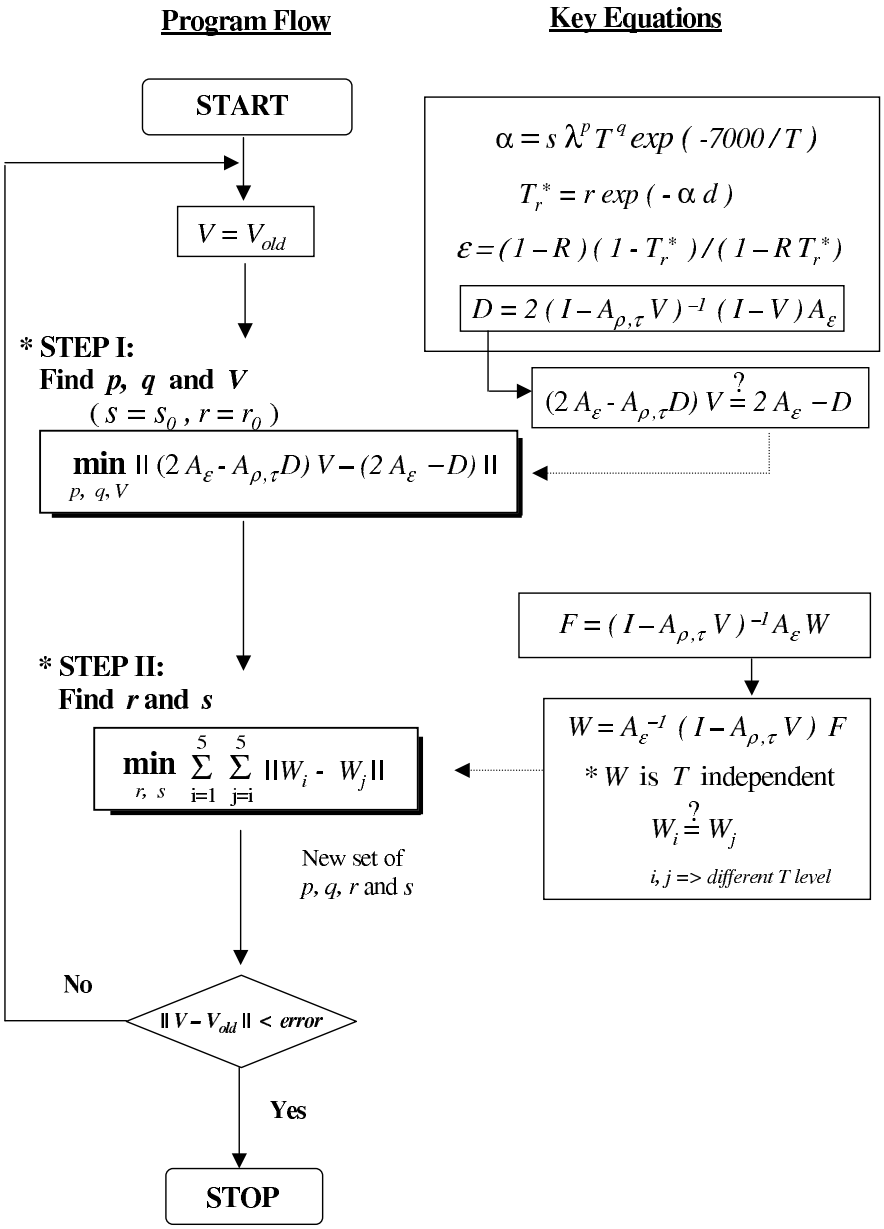


Figure 3.3: Parameter estimation program flow chart

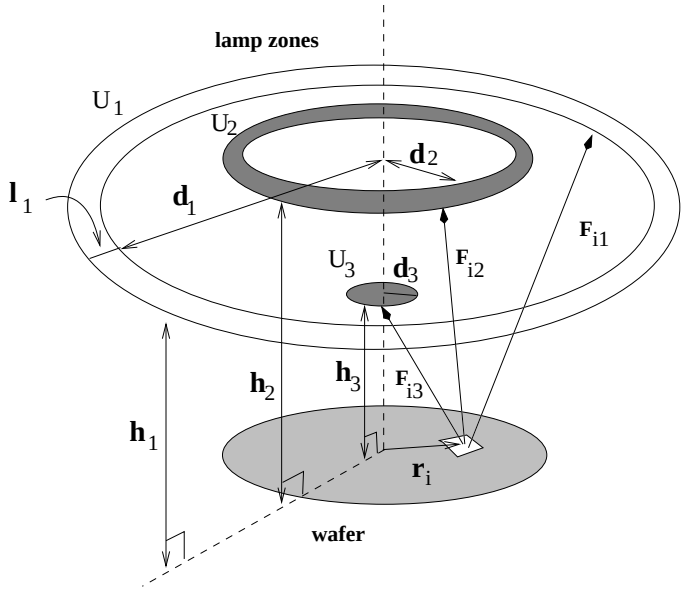


Figure 3.4: Schematic of view factors between an element on wafer and various lamp zones: U_1, U_2 and U_3 are lamp zones, r_i is the radial distance from center of radius to the ith element area of wafer, d_1 and d_2 are inner radii of lamp zones U_1 and U_2, respectively, d_3 is the radius of zone 3 lamp, h_1, h_2 and h_3 are heights of each lamp zones and F_{i1}, F_{i2} and F_{i3} are view factors from r_i location of the wafer surface to each lamp zones

The view factor between the wafer surface to lamp 3 is expressed as

$$F_3(r) = \frac{1}{2}\left[1 - \frac{r^2 + h^2 - d^2}{\sqrt{[(r+d)^2 + h^2][(r-d)^2 + h^2]}}\right] \tag{3.46}$$

where r is the radial distance of a point on a wafer, h is the height of lamp zone from wafer surface, and d is the radius of the lamp disk.

The view factors between the wafer surface and to lamp zone 1 and 2 can

Table 3.1: Parameter values for view factor calculation

Lamp zone	h (inch)	d (inch)	l (inch)
1 (Edge)	2.0	3.84	0.3125
2 (Middle)	3.0	2.22	0.3125
3 (Center)	1.75	0.1875	N/A

be expressed as

$$F_i(r) = \frac{1}{2}\frac{r^2+h^2-d^2}{\sqrt{[(r+d)^2+h^2][(r-d)^2+h^2]}} - \frac{1}{2}\frac{r^2+h^2-(d+l)^2}{\sqrt{[(r+d+l)^2+h^2][(r-d-l)^2+h^2]}} \tag{3.47}$$

where i is either 1 or 2, and d is the inner radius of the lamp ring.

The physical dimension of the necessary variables, r, h, d and l are shown in Figure 2.7 of Chapter 2 and actual parameter values for h, d and l of each zone are listed in Table 3.1. View factors from a wafer element area to each lamp zone(zones 1,2 and 3) can be plotted as a function of radial distance of the element area from the center of wafer and is shown in Figure 3.5.

3.6 Numerical Modeling

Equation 3.25 does not have an analytical solution. At steady state, with an assumption of uniform heat input Q(constant Q per unit area of wafer surface per unit time), Bentini et al.[6] showed the solution for the simplified model equation with modified Bessel function. But the design of the uniformly irradiating heat source is almost impossible. And the solution by Bentini et al.[6] shows temperature non-uniformity with lower temperature at the edge. Most industrial rapid thermal

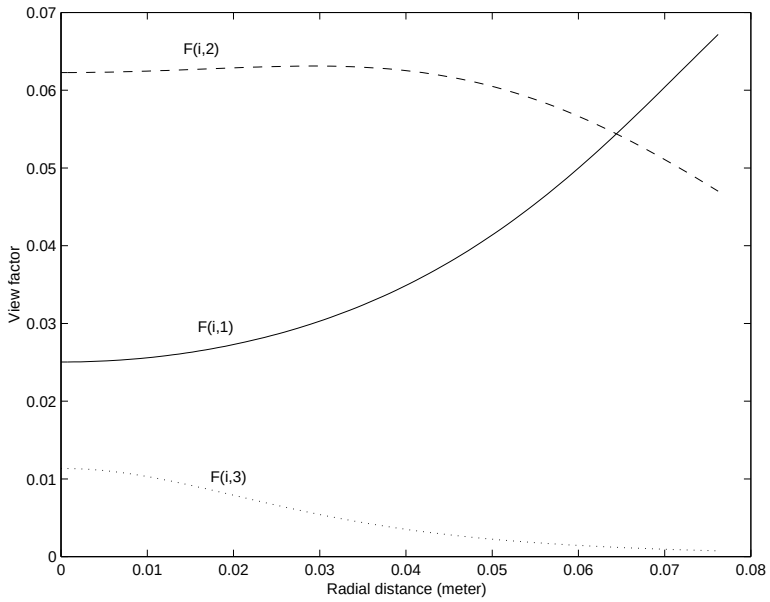


Figure 3.5: Plot of view factors as a function of radial distance

processors are designed to have multiple heat sources with independent control signal to efficiently compensate the non-uniform temperature profile. For multiple heat source reactors, view factors and heat radiation terms are nonlinear functions of r and T (with T^4), respectively. Some industrial reactors have multiple lamps with single control, but experiments [25] show severe temperature non-uniformity at steady state which may not be easily explained by the Bessel function solution.

In this research, the solution to equation 3.25 was obtained numerically. The forward time centered space (FTCS)scheme[27] is applied to obtain the dynamics of the system. Since the heat conduction terms are not left out, dynamic interactions due to heat conduction between wafer zones as well as due to radiations can be

simulated.

When the wafer radius R is divided by N equally spaced grid of length $h(= R/N)$ and τ is the time step, temperature T at radial position r and time t is expressed as $T(r,t) = T_i^n$, where i is from $r_i = ih$ and n is from $t = n\tau$. Then

$$\frac{\partial T}{\partial t} = \frac{T_i^{n+1} - T_i^n}{\tau}$$

$$\frac{\partial T}{\partial r} = \frac{T_i^n - T_{i-1}^n}{h}$$

$$\frac{\partial^2 T}{\partial^2 r} = \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{h^2}$$

Equation 3.25 can be rearranged as

$$\begin{aligned} Z\rho C_p \frac{dT_i}{dt} &= kZ\left(\frac{1}{r}\frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial r^2}\right) \\ &\quad -\sigma D T_i^4 + FP - h_w(T_i - T_a) \end{aligned} \tag{3.48}$$

And the final discretized form of equation 3.25 is

$$\begin{aligned} T_i^{n+1} &= \frac{k\tau}{\rho C_p}\left[\frac{1}{ih^2}\left(iT_{i+1}^n + (1-2i)T_i^n + (i-1)T_{i-1}^n\right)\right] \\ &\quad + \frac{\tau}{Z\rho C_p}\left[\sum_{j=1}^{3} F_{ij}P_j - \sigma D_{i,sum}T_i^{n4} - h_w(T_i - T_a)\right] + T_i^n \end{aligned} \tag{3.49}$$

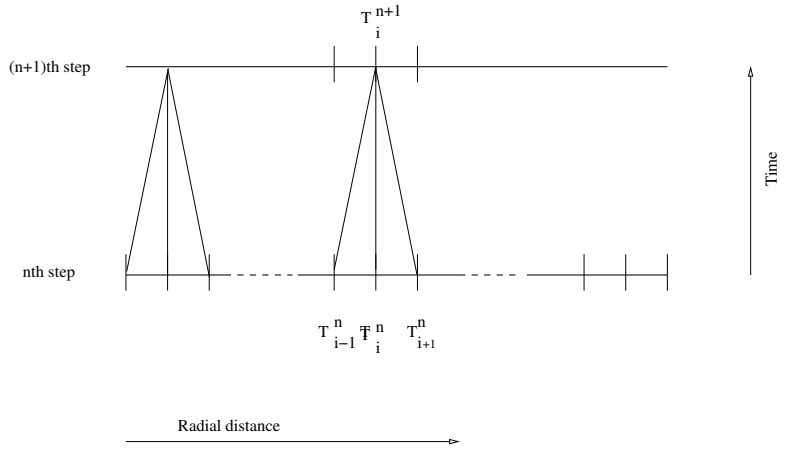


Figure 3.6: Relationship between n^{th} and $n+1^{th}$ step calculation

3.7 Conclusions

Process gain K_{ij} and τ_i can be obtained from experiments. These parameters (equations 3.36 and 3.37) are derived from the heat balance equation and are functions of effective view factor F_{ij}, effective emissivity D_i, temperature and convective heat transfer coefficient. Since F_{ij} and D_{ij} are also dependent on the wafer total emissivity A_ϵ, wafer zone to zone view factor V, and lamp to wafer view factor W, these parameter value estimation sequence was derived. Consequently sensitivity of process gain to temperature $\partial K_{ij}/\partial T_i$ (equation 3.41), and time constant to temperature $\partial \tau_i/\partial T_i$ (equation 3.45) of RTP chamber can be explicitly represented from a set of identification experiments described in Chapter 4.

Chapter 4

Identification of Wafer Temperature Dynamics

As briefly mentioned in Chapter 1, open-loop identification of the temperature dynamics which utilizes temperature responses to step input responses of lamp power can be less effective than closed-loop identification. Namely advantages of closed-loop control over open-loop control can be explained in more detail as follows

- Wafer temperature is kept under control: In open-loop method, since there is no feedback for the loop being identified, the temperature increase can be unexpectedly large. This may damage items of RTP chamber including wafer holder mechanisms or quartz window (if one part of quartz plate becomes to hot compared to other part, since the edge of plate is fixed, the thermal stress may break the plate).

- Data collection time (identification time) can be reduced: For closed-loop identification, we can preset the identification time to same level for each identification step.

- Detrending step of drift data from response data is not needed. The disturbance l_i can be described as a ramp function as explained in section 4.2.2 and the amount (slope) of disturbance can be identified from equations 4.11 and 4.13.

- Required lamp power levels to obtain the same temperature changes in each set point can be automatically applied. In an open-loop step test, one may apply a step change to lamp power without knowing a priori what temperature level response one would get for the output temperature.

4.1 Open-loop Identification through Input Step Changes

The 3 x 3 MIMO transfer function matrix can be generally expressed as

$$G(s) = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}$$

where g_{ij} denotes $g_{ij}(s)$. In this work, each element transfer function g_{ij} is assumed to be first order plus time delay(θ_{ij}). l_i's, denote the temperature drift transfer functions to be added to the ith output variable. Then the transfer function g_{ij} is

$$\begin{aligned} g_{ij} &= \frac{K_{ij}e^{-\theta_{ij}s}}{\tau_{ij}s+1} \\ &= \frac{n_{ij}}{d_{ij}} \end{aligned} \tag{4.1}$$

We omit the s argument from each transfer function for simplicity. When a step change in input of size b_j is applied to g_{ij} without feedback and the disturbance l_i is a ramp function with slope α_i, the output y_{ij} is

$$\begin{aligned} y_{ij} &= g_{ij} \cdot u_j + l_i \\ &= \frac{K_{ij}e^{-\theta_{ij}s}}{\tau_{ij}s+1} \cdot \frac{b_j}{s} + \frac{\alpha_i}{s^2} \end{aligned} \tag{4.2}$$

In order to estimate the parameters from the sampled data set, equation 4.2 is transformed to discrete form using the modified z-transformation technique. When

the delay time $\theta_{ij}^{'}$ of the element transfer function g_{ij} is expressed as

$$\theta_{ij} = N_{\theta ij}T + \theta_{ij}{}^{'}$$

where $N_{\theta ij}$ is an integer and $0 < \theta_{ij}{}^{'} < T$, and T is the sampling time. Using the modified z-transformation with zero order hold results in following pulse transfer function

$$\begin{aligned} hg_{ij}^{*} &= z^{-(N_{\theta ij}+1)} \frac{K_{ij}\{(1-e^{-m_{ij}T/\tau_{ij}}) - (e^{-T/\tau_{ij}} - e^{-m_{ij}T/\tau_{ij}})z^{-1}}{(1-e^{-T/\tau_{ij}}z^{-1})} \\ &= z^{-(N_{\theta ij}+1)} \frac{n_{ij}^{*}}{d_{ij}^{*}}. \end{aligned} \tag{4.3}$$

where $m_{ij} = 1 - \theta_{ij}{}^{'}/T$. Thus

$$y_{ij}^{*} = z^{-(N_{\theta ij}+1)} \frac{n_{ij}^{*}}{d_{ij}^{*}} y_{s_j}^{*} + l_{i}^{*} \tag{4.4}$$

where $y_{s_j}^{*}$ is a step change, b_j and l_i^{*} is a ramp change of slope α_i in sampled data system.

4.2 MIMO Closed-loop Identification

4.2.1 Continuous systems

The block diagram for the multiloop SISO control for this input/output transfer function system is shown in Fig. 4.1. The controller used is a proportional plus integral(PI) controller. The controller g_{c_i} for a PI controller can be expressed

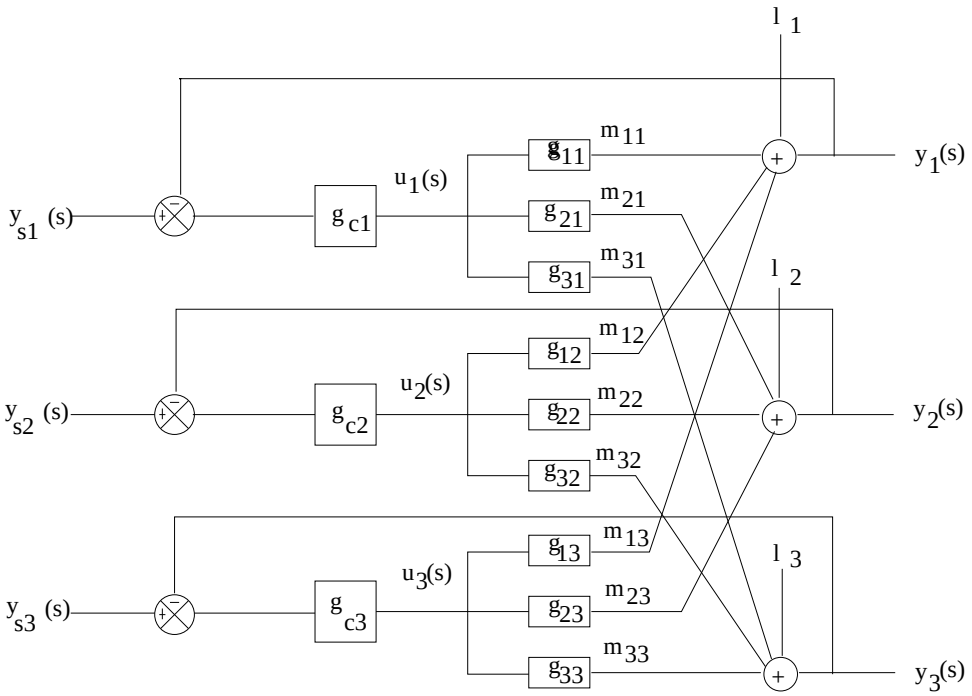


Figure 4.1: Block diagram of 3x3 multi-loop PID control system(continuous case)

as

$$\begin{aligned} g_{c_i} &= K_{c_i}(1 + \frac{1}{\tau_{I_i} s}) \\ &= \frac{K_{c_i}(\tau_{I_i} s + 1)}{\tau_{I_i} s} \\ &= \frac{n_{c_i}}{d_{ci}} \end{aligned} \tag{4.5}$$

Let y_{ij} be the closed-loop response of ith output variable when the control loops other than jth loop are open(controller outputs of ith($i \neq j$) loops are held at constant values). The closed loop response of y_{ij} to the step changes in set point

y_{s_j} can be expressed as

$$y_{ij} = l_i + \frac{g_{ij} g_{c_j} (y_{s_j} - l_j)}{1 + g_{jj} g_{c_j}} \tag{4.6}$$

when $i = j$, equation 4.6 reduces to

$$y_{jj} = \frac{g_{jj} g_{c_j} y_{s_j} + l_j}{1 + g_{ij} g_{c_j}} \tag{4.7}$$

Substituting equations 4.1 and 4.5 into equations 4.6 and 4.7 results in

$$y_{ij} = l_i + \frac{n_{ij} n_{c_j} d_{jj} (y_{s_j} - l_j)}{d_{ij} (d_{jj} d_{cj} + n_{jj} n_{c_j})} \tag{4.8}$$

$$y_{jj} = \frac{n_{jj} n_{c_j} y_{s_j} + d_{jj} d_{cj} l_j}{d_{jj} d_{cj} + n_{jj} n_{c_j}} \tag{4.9}$$

In this research the closed-loop identifications with PI control are utilized. Actual closed-loop estimation results are listed and analyzed. The set-point changes(y_{s_j}'s) are step changes with magnitude of b_j's and the temperature drift disturbances(l_j's) are assumed to be a ramp function with slopes α_j's. i.e., in frequency domain,

$$y_{s_j}(s) = \frac{b_j}{s}$$

$$l_j(s) = \frac{\alpha_j}{s^2}$$

4.2.2 PI controller case-continuous system

When the controller is in PI mode and $i = j$, substituting the process and the controller parameters to equation 4.9 gives

$$y_{jj} = \frac{\overbrace{K_{jj} e^{-\theta_{jj} s} K_{c_j} (\tau_{Ij} s + 1) \cdot \frac{b_j}{s}}^{\text{I}} + \overbrace{(\tau_{jj} s + 1) \tau_{Ij} s \cdot \frac{\alpha_j}{s^2}}^{\text{II}}}{\tau_{jj} \tau_{Ij} s^2 + (\tau_{Ij} + K_{jj} e^{-\theta_{jj} s} K_{c_j} \tau_{Ij}) s + K_{jj} e^{-\theta_{jj} s} K_{c_j}} \tag{4.10}$$

where part I and II in the numerator are related to the step change in set point, and the ramp change in the temperature drift disturbance, respectively. We define the long-time behaviour of y_{jj} as $\bar{y}_{jj}$, which is

$$\bar{y}_{jj} = b_j + \frac{\tau_{Ij}}{K_{jj}K_{c_j}}\alpha_j \tag{4.11}$$

When $i \neq j$, substituting the process and the controller parameters in equation 4.8 results in

$$y_{ij} = l_i + \frac{K_{ij}e^{-\theta_{ij}s}K_{c_j}\{\tau_{jj}\tau_{Ij}s^2 + (\tau_{jj}+\tau_{Ij})s + 1\}(\overbrace{\frac{b_j}{s}}^{\text{I}} - \overbrace{\frac{\alpha_j}{s^2}}^{\text{II}})}{(\tau_{ij}s+1)\{\tau_{jj}\tau_{Ij}s^2 + \tau_{Ij}(K_{jj}e^{-\theta_{jj}s}K_{c_j}+1)s + K_{jj}e^{-\theta_{jj}s}K_{c_j}\}} \tag{4.12}$$

The long-time behavior for the response from the step change in set point(part I) can still be obtained as $K_{ij}b_j/K_{jj}$. After partial fraction expansion and finding the coefficients for repeated factors, one can obtain the asymptote due to the ramp change(part II) as

$$(\frac{\tau_{Ij}}{K_{jj}K_{c_j}} + \tau_{ij} - \tau_{jj})\frac{K_{ij}}{K_{jj}}\alpha_j - \frac{K_{ij}}{K_{jj}}\alpha_j t$$

Thus the long-time behavior is

$$\bar{y}_{ij} = \frac{K_{ij}}{K_{jj}}b_j + (\frac{\tau_{Ij}}{K_{jj}K_{c_j}} + \tau_{ij} - \tau_{jj})\frac{K_{ij}}{K_{jj}}\alpha_j + (\alpha_i - \frac{K_{ij}}{K_{jj}}\alpha_j)t \tag{4.13}$$

Equation 4.13 reduces to equation 4.11 when $i = j$.

4.2.3 P Controller case-continuous system

When the controller is in P mode and $i = j$, following same procedure as in PI mode, equation 4.9 becomes

$$y_{jj} = \frac{\overbrace{K_{jj}K_{c_j} \cdot \frac{b_j}{s}}^{\text{I}} + \overbrace{(\tau_{jj}s + 1) \cdot \frac{\alpha_j}{s^2}}^{\text{II}}}{\tau_{jj}s + 1 + K_{jj}K_{c_j}} \tag{4.14}$$

The same procedure applied to equation 4.12 is used to obtain the asymptote

$$\bar{y}_{jj} = \frac{K_{jj}K_{c_j}}{1 + K_{jj}K_{c_j}} b_j + \frac{K_{jj}K_{c_j}\tau_{jj}}{(1 + K_{jj}K_{c_j})^2}\alpha_j + \frac{1}{1 + K_{jj}K_{c_j}}\alpha_j t. \tag{4.15}$$

When $i \neq j$, following the similar procedure as above,

$$y_{ij} = l_i + \frac{K_{ij}e^{-\theta_{ij}s}K_{c_j}(\tau_{jj}s + 1)(y_{s_j} - l_j)}{(\tau_{ij}s + 1)(\tau_{jj}s + 1 + K_{jj}e^{-\theta_{jj}s}K_{c_j})} \tag{4.16}$$

$$\begin{aligned} \bar{y}_{ij} &= \frac{K_{ij}K_{c_j}}{1 + K_{jj}K_{c_j}} b_j + \frac{K_{ij}K_{c_j}\{\tau_{ij} + (\tau_{ij} - \tau_{jj})K_{jj}K_{c_j}\}}{(1 + K_{jj}K_{c_j})^2}\alpha_j \\ &\quad + (\alpha_i - \frac{K_{ij}K_{c_j}}{1 + K_{jj}K_{c_j}}\alpha_j)t \end{aligned} \tag{4.17}$$

Equation 4.17 reduces to equation 4.15 when $i = j$.

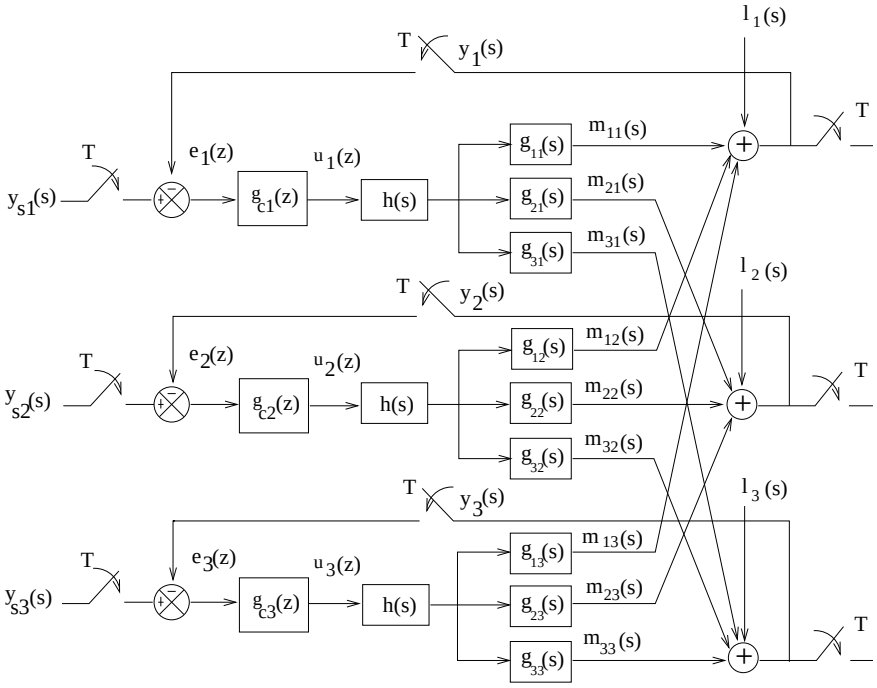


Figure 4.2: Block diagram of 3x3 multi-loop PID control system(sampled data case)

4.3 MIMO Closed-loop Identification–Sampled Data System

Refer to Figure 4.2 for the block diagram of the sampled data system for a 3 x 3 multi-loop SISO control system.

A similar procedure as for continuous systems is used to derive the sampled data response behavior of the system. The closed-loop transfer functions are expressed as

$$y^*_{ij} = l^*_i + \frac{h g^*_{ij} g^*_{c_j} (y^*_{s_j} - l^*_j)}{1 + h g^*_{jj} g^*_{c_j}} \tag{4.18}$$

for $i = j$,

$$y^*_{jj} = \frac{hg^*_{jj}g^*_{c_j}y^*_{s_j} + l^*_j}{1 + hg^*_{jj}g^*_{c_j}} \tag{4.19}$$

where f^* or $f^*(z)$ denote the z-transformation of the function f. Substituting equation 4.3 into 4.18 and 4.19 gives

$$y^*_{ij} = l^*_i + \frac{z^{-(N_{\theta ij}+1)}d^*_{jj}n^*_{ij}n^*_{c_j}(y^*_{s_j} - l^*_j)}{d^*_{ij}d^*_{jj}d^*_{cj} + z^{-(N_{\theta ij}+1)}d^*_{ij}n^*_{jj}n^*_{c_j}} \tag{4.20}$$

$$y^*_{jj} = \frac{z^{-(N_{\theta jj}+1)}n^*_{jj}n^*_{c_i}y^*_{s_i} + d^*_{jj}d^*_{ci}l^*_i}{d^*_{jj}d^*_{ci} + z^{-(N_{\theta jj}+1)}n^*_{jj}n^*_{c_i}}. \tag{4.21}$$

4.3.1 PI and P controllers

For a PI controller,

$$\begin{aligned} g^*_{c_j} &= K_{c_j}(1 + \frac{T}{\tau_{Ij}}\frac{1}{1 - z^{-1}}) \\ &= \frac{K_{c_j}\{(1 + T/\tau_{Ij}) - z^{-1}\}}{1 - z^{-1}} \end{aligned} \tag{4.22}$$

And for P controller,

$$g^*_{c_j} = K_{c_j} \tag{4.23}$$

4.4 Program Flow of the Closed-loop Identification

Each identification method, i.e., open-loop, closed-loop PI and P control identification methods introduced in previous sections, has been implemented with

MATLAB®. Constrained parameter estimation was performed using successive quadratic optimization with sum of the squared error between model and the experimental response as objective function. Time delay(θ_{ij}), gain(K_{ij}), time constant(τ_{ij}) and temperature drift slope(α_j) were the estimation parameters. The MATLAB routine constr was used for the optimization.

The model were expressed as a discrete transfer function which was represented as coefficients in numerator and denominator, namely, equations 4.3, 4.20 and 4.21. MATLAB routine dstep is used to obtain the output responses.

4.5 Closed-loop Identification Experiment

Identification experiments were performed for 5 different temperature levels of 300°C , 400°C , 500°C , 600°C , and 700°C . The temperature response curves for each temperature level identification experiment are plotted in Figure 4.3. The wafer temperature set point was set as 100°C at the beginning and the set point was ramped to the final set point for identification. Figure 4.4 shows data points for 600°C identification experiment in detail. Detailed view of temperature responses during three identification steps at around 600°C , overall view of temperature responses and controller output levels are shown in Figure 4.4 (a), (b) and (c), respectively.

As described in Chapter 2, ith control loop, controls wafer temperature T_i, with manipulated variable U_i (ith lamp zone). For each temperature level, after the temperature response was settled to identification temperature level, ith output

variable set point was increased by specified amount (ΔT_s, mostly 5°C) while the other loop is kept at the manual mode. In manual mode, the controller output of that loop is kept constant at the same level as when the identification step started. For example, when the set point is increased for identification for first loop (T_1 and U_1 pair), PID controllers' operation mode for loops 2 and 3 are switched from auto modes to manual modes which keeps controller output variables U_2 and U_3 at constant level when they were switched. After identification for ith loop, set points of all three PID loops are set to same final set points, 600°C for this example, and the PID controllers' operation modes are changed from manual to auto mode. After the temperature responses settle down to the final set points, identification experiments for 2nd and 3rd loops are performed sequentially.

4.5.1 Identification results: K_{ij} and τ_i

Using the MATLAB routines described in section 4.4, K_{ij} and τ_i values from each identification experiments were estimated and plotted in Figures 4.5 and 4.6, respectively. View factors F_{ij} are estimated from τ_i and K_{ij} and plotted in Figure 4.7.

$h_{w,i}$ is estimated from cooling curve data and the value is 11 Watts/(m^2K). The τ_i values are obtained from identification experiments and $D_{i,sum}$ are estimated from equation 3.37. F_{ij} derived in Chapter 3 can be estimated from equation 3.38 using the parameter values identified from the experiments. Then these F_{ij} values can be used to estimate lamp to wafer view factor W_{ik}, which will be used for numerical modeling of the wafer heat balance equation of RTP chamber. Then using the

estimation steps described in Chapter 3 parameters needed for process gain and time constant sensitivity to temperature can be obtained.

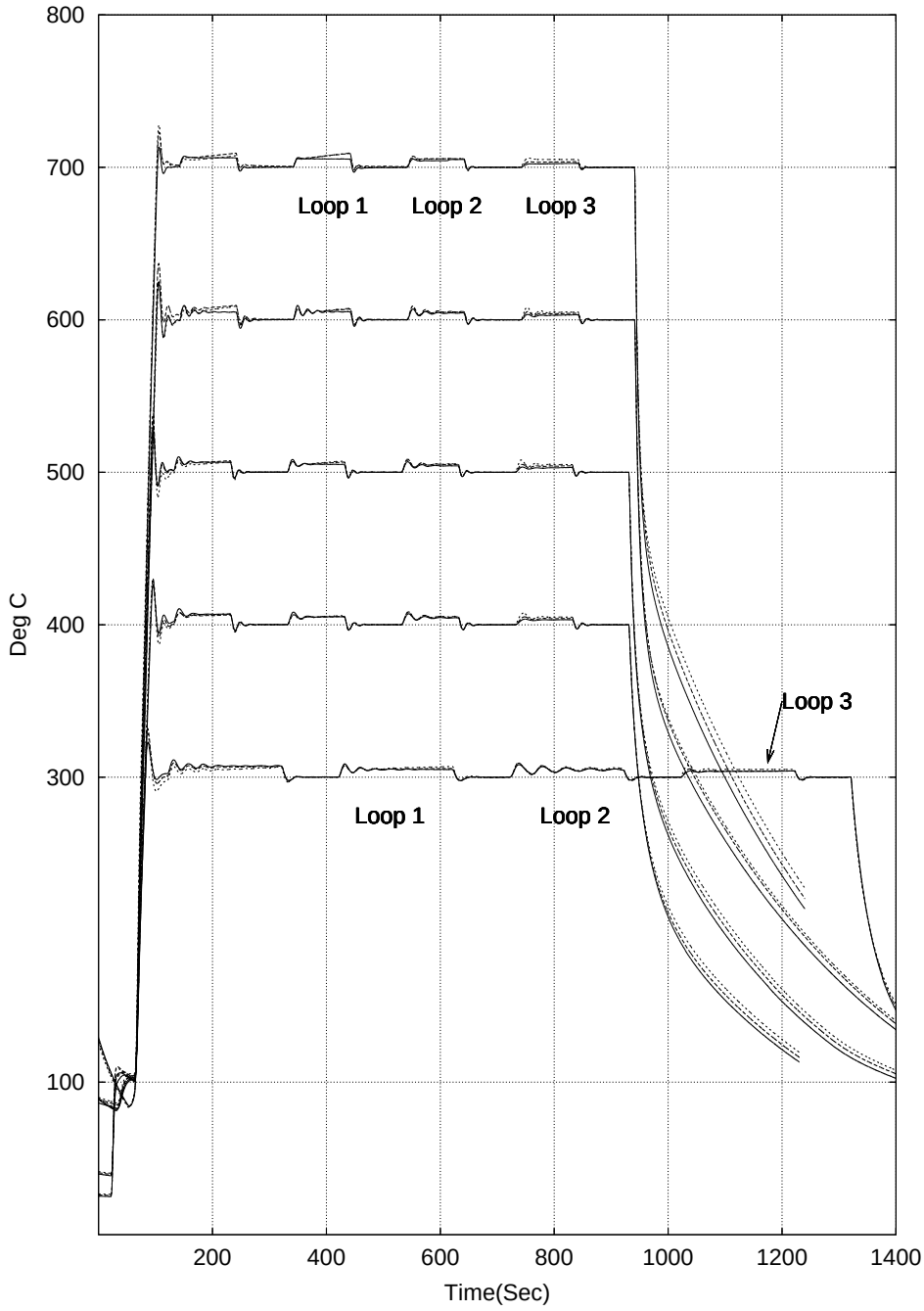


Figure 4.3: Overall view of the temperature responses for identification experiments: 5 experiments are shown for 300°C , 400°C , 500°C , 600°C and 700°C experiments. Identification time for 300°C level was 200 seconds and those of other higher temperature levels were 100 seconds

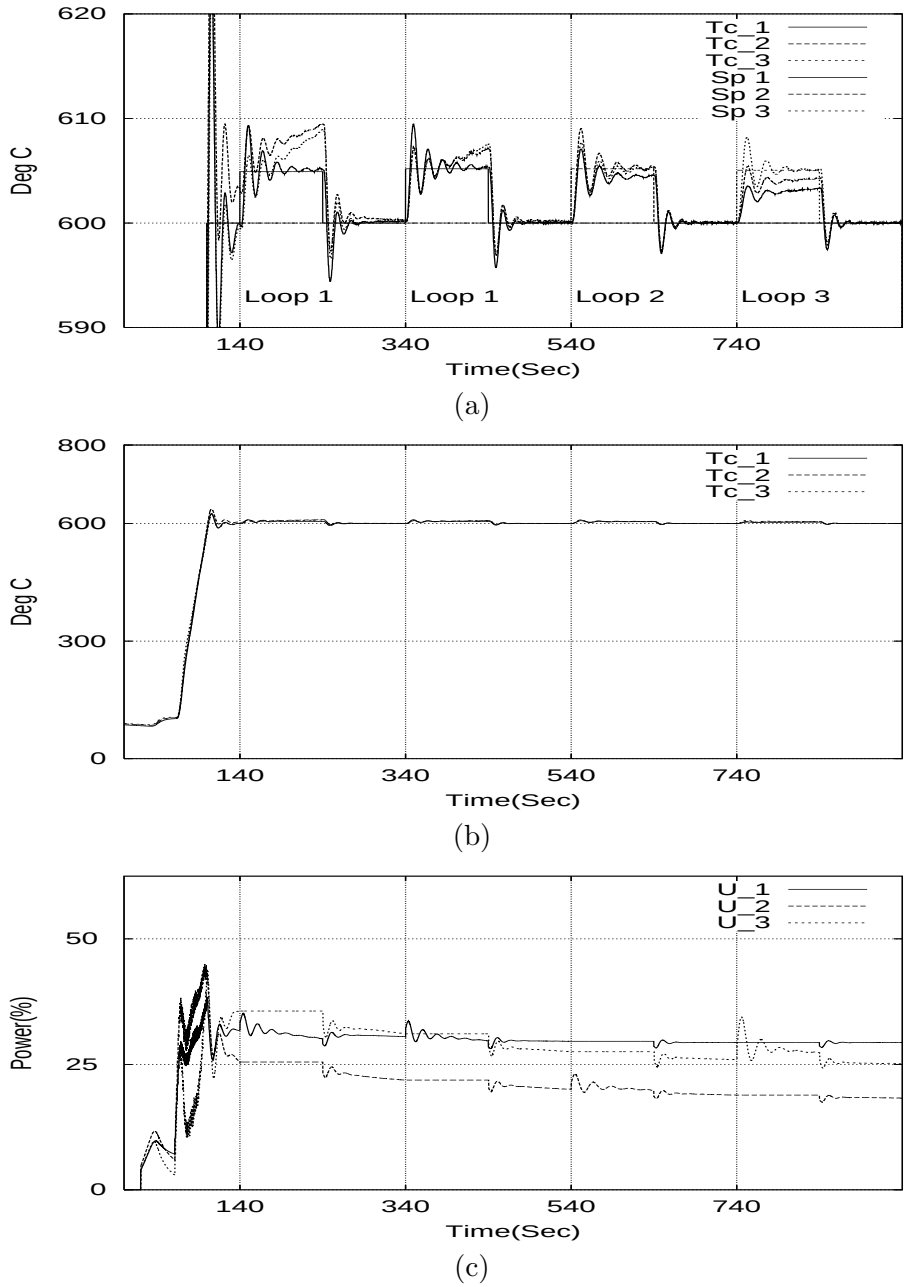


Figure 4.4: (a) Detail responses for 600 °C identification, (b) Overall responses for 600 °C identification run, (c) Controller output changes

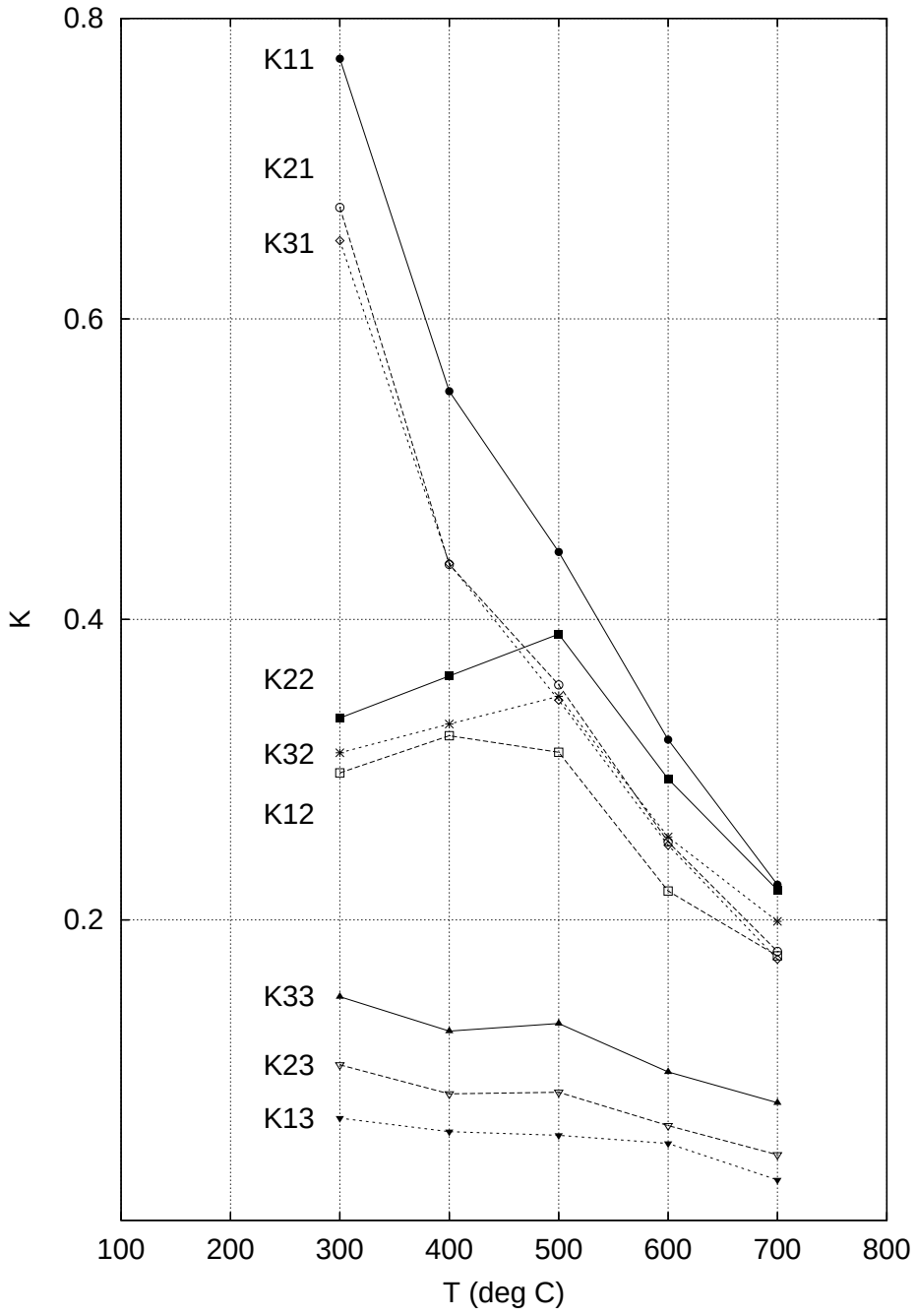


Figure 4.5: K_{ij} vs temperature

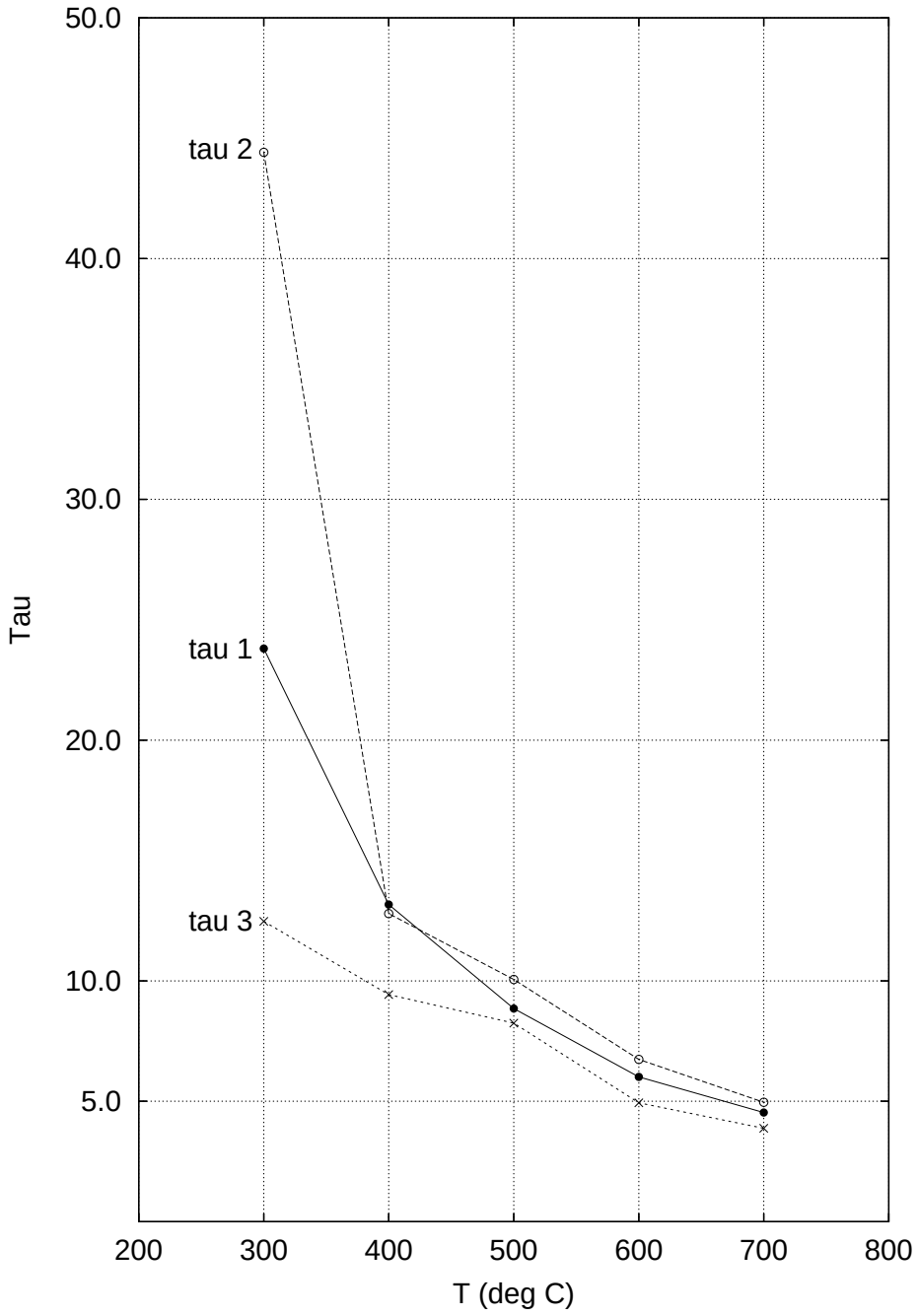


Figure 4.6: τ_i vs temperature plots

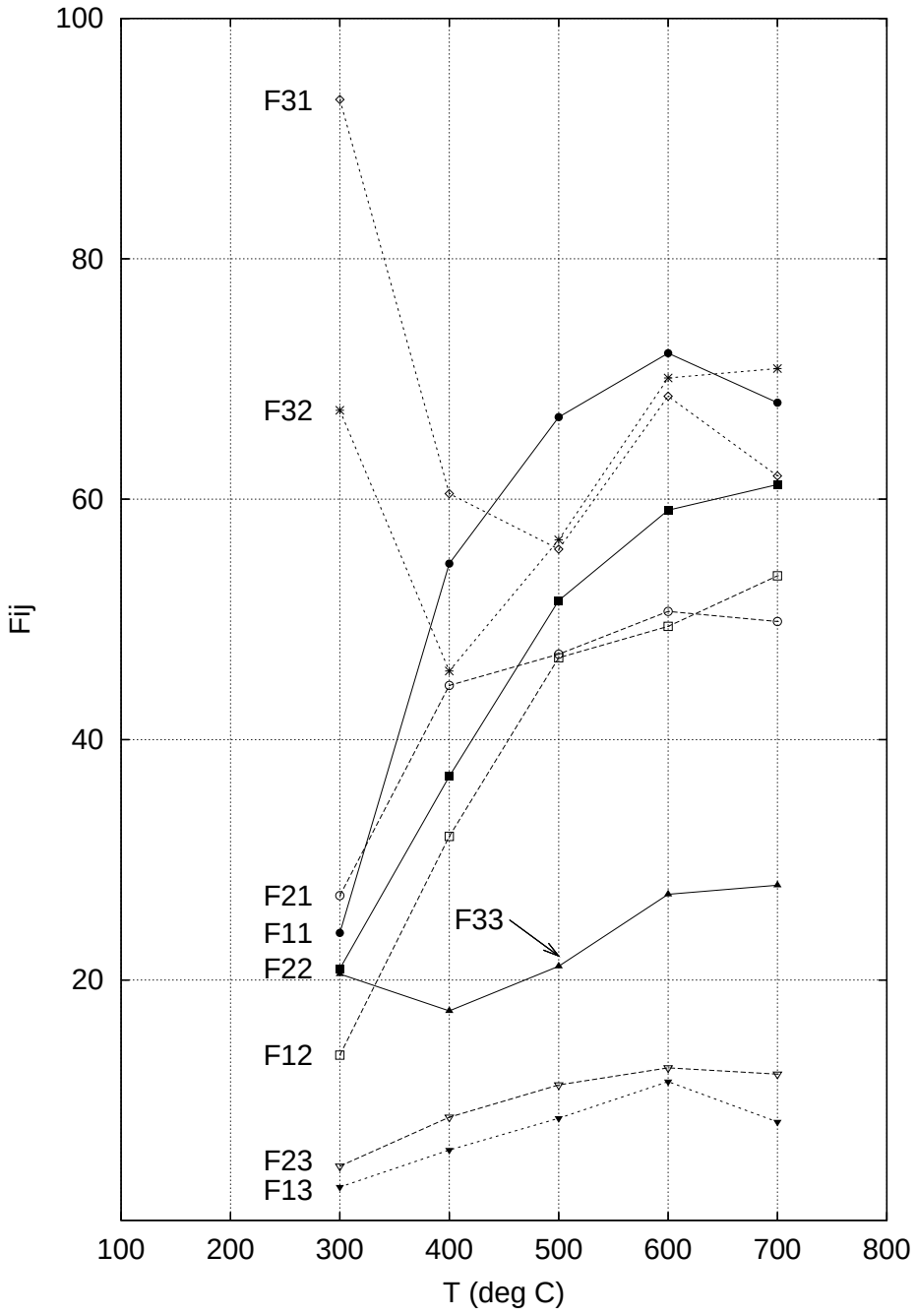


Figure 4.7: F_{ij} vs temperature plots

Chapter 5

Iterative Identification Methods for Ill-Conditioned Processes

The condition number for the RTP system matrix is typically large and considered to be ill-conditioned for control purposes. Some ill-conditioned processes are very sensitive to small element wise uncertainties arising in classical element-by-element model identification. For such processes, accurate identification of singular values and right singular vectors are more important than those of elements themselves. Singular values and right singular vectors can be found by iterative identification methods that implement the input and output transformation iteratively. Methods based on SVD decomposition, QR decomposition, and LU decomposition are proposed and compared with Koung and MacGregor's method. Convergence proofs are given. These SVD and QR methods use orthogonal matrices for the transformations that cannot be calculated analytically in general, and so they are hard to apply to dynamic processes, whereas the LU method uses simple analytic transformations and can be directly applied to dynamic processes.

5.1 Introduction

Interaction between methods of identification and control is a recent topic of interest. Classical identification approaches often do not provide information

appropriate for robust controller design methods. Control relevant identification methods that will give better models to design control systems are required (Gevers[29]). These include identification of uncertainty regions in addition to the nominal model, worst case identifications matching the framework of H_∞ control, and iterative closed-loop identification leading to successively better design of controllers (Koung and MacGregor, [45])

For ill-conditioned multivariable processes, additional attention should be paid to identification. Inversions of some ill-conditioned processes are very sensitive to modeling error. That is, small element wise error arising in the classical identification can be amplified. Explicit and implicit model-based control systems that use inversion can exhibit poor robust performances. In such cases, it is very important to accurately determine the small singular values that become large in inversion and the corresponding singular vectors (Li and Lee[52]).

Usually, equal input perturbations do not excite the small singular values and their singular vectors sufficiently. Anderson et al.[2] have shown that closed-loop identification can excite the small singular values and their corresponding singular vectors more completely than classical open-loop identification methods and provide a better model to design control systems. With input transformations via right singular vectors assumed to be known a priori, Koung and MacGregor [44] have proposed an identification method that can excite each singular value and its singular vector separately. Later, they also proposed an iterative method that refines the right singular vectors (Koung and MacGregor [45]).

Li and Lee [52] have proposed an identification method for an ill-conditioned

multivariable process that fits both the process model and its inverse. They found the inverse of the process from an experimentally determined relative gain array (Seborg et al.[77]). Achieving a small error in the process gain matrix guarantees accurate large singular values and singular vectors. To obtain the relative gain array experimentally, stable control systems with integral action must be designed, which may not be easy for higher order processes.

Ill-conditioned processes consist of large elements due to large singular values and small elements due to small singular values. For the purpose of control system design, it is more important to determine accurately the small elements due to small singular values. The small elements cannot be found accurately unless they are separated from the large elements. Components due to large singular values and small singular values should be separated and then be identified independently. Koung and MacGregor's [44] method can achieve this when the right singular vectors are known. They also proposed an iterative method [45] that can be applied for poorly known right singular vectors. However, a convergence proof has not been established yet for higher order processes. In this paper, three iterative identification methods for ill-conditioned multivariable processes are proposed and compared with the Koung and MacGregor method [45]. Convergence proofs are given with small identification error at each step. Properties of each method are explained.

5.2 Robust Models for Designing Control Systems

Classical SISO identification usually does not provide models adequate to design control systems for ill-conditioned processes. However, not all ill-conditioned

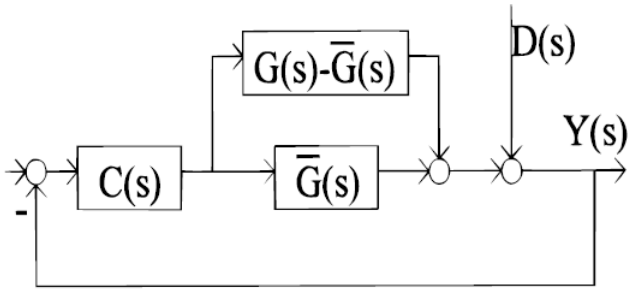


Figure 5.1: Schematic of a feedback control system

processes are sensitive to the model identification error. In this section, we propose measures to know whether a new iterative identification is needed.

Consider the multivariable control system in Figure 5.1. The transfer function matrix between the disturbances and the outputs is

$$Y(s) = [I + G(s)C(s)]^{-1}D(s) \tag{5.1}$$

When the process model $\bar{G}(s)$ is used for designing the control system, it becomes

$$\begin{aligned} Y(s) &= [I + G(s)C(s)]^{-1}D(s) \\ &= [I + \bar{G}(s)C(s)]^{-1}[I + E(s)H(s)]D(s) \end{aligned} \tag{5.2}$$

where

$$E(s) = \bar{G}(s)G^{-1}(s) - IH(s) = [I + G(s)C(s)]^{-1}G(s)C(s)$$

Here we only consider the steady state gain matrix and omit the argument s below. At the steady state, the complementary sensitivity function $H(s)$ is usually

identity because integral action is used for offset free operation. Hence if the term (Grosdidier and Morari [34])

$$E = \bar{G}G^{-1} - I \tag{5.3}$$

is small, the performance of the control system 5.1 would not be much different from what is expected at the design stage. That is, for robust performance, a model with small $||\bar{G}G^{-1} - I||$ is recommended (Koung and MacGregor [45]) and, for robust stability, a model with small $\min_{D=\text{diagonal}}||D^{-1}(\bar{G}G^{-1} - I)D||$ is recommended because $\det(\mathrm{I} + \mathrm{E}) = \det(\mathrm{I} + \mathrm{D}^{-1}\mathrm{ED})$.

When the process gain matrix G is identified element-by-element with a certain SISO identification method, it will be reasonable to assume that each element has a small relative error. mathematically it can be represented by

$$\hat{G} = G + \epsilon G \circ \Delta \tag{5.4}$$

where $\circ$ means the Hadamard product (element by element product) and $\epsilon\Delta$ is a relative element error with a small number ϵ and $||\Delta|| = 1$. For the identification error model of equation 5.4, various condition numbers have been defined in Lee et al.[48]. The following uncertainty condition numbers are used to quantify the model uncertainty effects:

(i) For robust performance,

$$\begin{aligned} \gamma_p(G) &\equiv \max_{||\epsilon\Delta||_p=1} ||\bar{G}G^{-1} - I||_p \\ &= \max_{||\epsilon\Delta||_p=1} ||(G \circ \Delta)G^{-1}||_p \\ &= ||(G^{-1} \otimes I)\mathrm{diag}(g_{11}, g_{21}, ..., g_{nn})||_{ip} \end{aligned} \tag{5.5}$$

(ii) For robust stability,

$$\gamma_p^* = \min_{\text{D=diagonal}} \gamma_p(DG) \tag{5.6}$$

where $||\cdot||_p$ and $||\cdot||_{ip}$ mean the matrix norm and the induced matrix norm, respectively, and $\otimes$ means the Kronecker product. Robust stability and robust performance can be obtained easily with the classical element by element identification, if $\gamma_p^*(G)$ and $\gamma_p(G)$ are small, respectively. The uncertainty condition number $\gamma_P(G)$ is input scaling invariant like the right minimally scaled condition number and $\gamma_P^*(G)$ is input and output scaling invariant like minimally scaled condition number and the relative gain array (Lee et al.[48]). That is,

$$\begin{aligned} \gamma_p(G) &= \gamma_p(GD) \approx k_R(G) \\ {\gamma_p}^*(G) &= {\gamma_p}^*(D_1 G D_2) \\ &\approx ||\Lambda = (G^{-T} \circ G|| \\ &\approx k^*(G) \end{aligned} \tag{5.7}$$

where $k_R(\cdot)$ and $k^*(\cdot)$ are the right minimally scaled condition number and minimally scaled condition number, respectively, and D, D_1, and D_2 are positive diagonal matrices. Instead of using a complex computation to find ${\gamma_p}^*(G)$, a norm of the RGA, $||\Lambda||$, can be used.

For some ill-conditioned processes, the two numbers, ${\gamma_P}^*(G)$ and $\gamma_P(G)$, can be very large and effects of the model identification error can be significant even for small identification error, making the element-by-element identification method

useless. To overcome this problem, Koung and MacGregor [44] have proposed an input transformation method with the right singular vector matrix. Actually, they identified the gain matrix of transformed process GV, where V is the right singular vector matrix. Since GV is identified, the identification error model can be

$$\bar{G}V = GV + \epsilon(GV \circ \Delta) \tag{5.8}$$

If the right singular vector matrix V is exact,

$$\begin{aligned} ||\bar{G}G^{-1} - I|| &= ||\epsilon(GV \circ \Delta)V^T G^{-1}|| \\ &= \epsilon||(W\Sigma \circ \Delta)\Sigma^{-1}W^T|| \\ &= \epsilon||(W \circ \Delta)W^T|| \\ &= O(\epsilon) \end{aligned} \tag{5.9}$$

where W is the left singular vector matrix. Since W is orthogonal ($W^T W = I$), the norm 5.9 is not large and there is no significant effect on the control performance and stability of the model identification error. When the right singular vectors are not known in advance, an iterative identification method refining the input transformation matrix can be used (Koung and MacGregor, [45]).

5.3 Iterative Identification Methods

5.3.1 Structure

The uncertainty condition numbers $\gamma_p(\cdot)$ and $\gamma_p^*(\cdot)$ are small for diagonal matrices and some triangular matrices. If the process can be converted to the diagonal (or triangular) form with appropriate transformations, robust models can

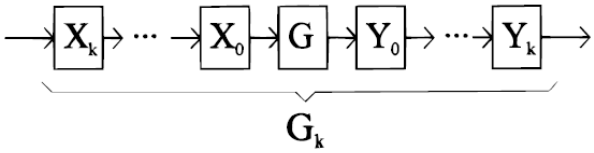


Figure 5.2: Input and output transformations for iterative identification

be obtained. Methods that diagonalize (or triangularize) the process successively with identified models having errors are investigated.

Transformations in the input and output are applied as shown in Figure 5.2. Let

$$G_k \equiv Y_k \cdots Y_0 G X_0 \cdots X_k \tag{5.10}$$

It can be identified as

$$\bar{G}_k = G_k + \epsilon G_k \circ \Delta_k \tag{5.11}$$

Then we have an identified model of the process gain matrix as

$$\bar{G} = Y^{-1}\bar{G}_k X^{-1} = G + \epsilon Y^{-1}(G_k \circ \Delta_k)X^{-1} \tag{5.12}$$

where $X = X_0 \cdots X_k$ and $Y = Y_k \cdots Y_0$. For this model,

$$\begin{aligned} ||\bar{G}G^{-1} - I||_P &= \epsilon||Y^{-1}(G_k \circ \Delta_k)G_k Y||_P \\ &\leq \epsilon\gamma_p(G_k)||Y||_p||Y^{-1}||_P||\Delta||_P \end{aligned} \tag{5.13}$$

Hence, if Y is well-conditioned and $\gamma_P(G_k)$ is small, the model $\bar{G}$ at the kth step will be adequate to design a control system having robust performance and robust stability.

Iterative methods that derive G_k to be diagonal (or triangular) with input and output transformations are considered. The procedure is as follows.

Step 1: Set $X_0 = Y_0 = I$ and identify G_0.

Step 2: Find X_1 and Y_1 such that $Y_1 G_0 X_1$ has a given diagonal (or triangular) structure and identify G_1.

Step 3: Repeat step 2 until the identified model is not much different from the given structure.

Conditions for convergence of the above iterative identification methods assuming a small identification error e at each step can be obtained: see [48] for details.

5.3.2 SVD method

The uncertainty condition number of Equation 5.5 is small for the diagonal matrices. The successive design of a full decoupler with $Y_k = I$ and $X_k = \bar{G}_k^{-1}$ is the simplest way for diagonalization. However, it is not practical because X_k is very sensitive to the model identification error and so the convergence radius of ϵ is usually small. The singular value decomposition (SVD) method can be used to diagonalize the process iteratively without sensitive transformations. Apply the singular value decomposition to the k^{th} estimates $\bar{G}_k$ as

$$\bar{G}_k = W_k \Sigma_k V_k^T \tag{5.14}$$

where W_k and V_k are both orthogonal matrices (that is, $W_k^T W_k = I$ and $V_k^T V_k = I$) and Σ_k is a positive diagonal matrix. Then

$$\begin{aligned} X_{k+1} &= V_k \\ Y_{k+1} &= W_k^T \end{aligned} \tag{5.15}$$

will diagonalize the process.

Let Θ be diagonal and $X = I + \epsilon^k A + O(\epsilon k + 1)$ and $Y = I + \epsilon^k B + O(\epsilon k + 1)$. Since X should be orthogonal such that

$$\begin{aligned} X^T X &= (I + \epsilon^k A^T + O(\epsilon^{k+1}))(I + \epsilon^k A + O(\epsilon^{k+1})) \\ &= I + \epsilon^k (A^T + A) + O(\epsilon^k + l) \\ &= I \end{aligned}$$

A is an antisymmetric matrix (that is, $A^T = -A$) and has $n(n-1)/2$ arbitrary elements. B is also an antisymmetric matrix and has the same arbitrary elements.

Since

$$\begin{aligned} Y(\Theta + \epsilon^k \tilde{\Theta})X &= (I + \epsilon^k B + O(\epsilon^{k+1})) \\ &\quad (\Theta + \epsilon^k \tilde{\Theta})(I + \epsilon^k A + O(\epsilon^{k+1})) \\ &= \Theta + \epsilon^k (B\Theta + \Theta A) + O(\epsilon^{k+l}) \end{aligned}$$

we can have a diagonal $B\Theta + \Theta A$ with appropriate A and B having $n(n-1)$ arbitrary elements. Applying this procedure to higher degrees of ϵ, we can obtain $Y(\Theta + \epsilon^k \tilde{\Theta})X \in \Omega$. This along with Lemma 1 in [48] proves convergence of this iterative method.

This method uses both the input transformation X_k and the output transformation Y_k. During the output transformation, arithmetic operations are applied to the measured output, which has a finite word length such as 12 bits. This may result in less accurate estimates.

5.3.3 QR method

Koung and MacGregor [45] have proposed an iterative identification method (KM method) that uses only the input transformation instead of both transformations of input and output:

$$\begin{aligned} X_{k+l} &= V_k \\ Y_{k+l} &= I \end{aligned} \tag{5.16}$$

Convergence of the KM method has not been proved yet. But simulations show that it is also convergent for a wide range of the relative identification error ϵ. An obstacle in proving the convergence of the KM method with Lemma 1 is that it does not transform the process to a given structure such as a diagonal and a triangular form. To overcome this, a slightly different method using QR decomposition has been derived. We decompose the kth estimates as

$$\bar{G}_k = P_k R_k^T Q_k^T \tag{5.17}$$

where P_k is a permutation matrix, Q_k is an orthogonal matrix, and R_k is an upper triangular matrix. Then

$$\begin{aligned} X_{k+l} &= Q_k \\ Y_{k+1} &= P_k^T \end{aligned} \tag{5.18}$$

will triangularize the process. For the triangular matrix, the uncertainty condition number 5.6 for stability is very low and the uncertainty condition number 5.5 for performance is usually much less than that of the original matrix.

The convergence of this method can be shown easily in a manner similar to the above SVD method.

5.3.4 LU method

The first three methods all use complicated transformations, and extension to dynamical systems may not be easy. A method that can be readily extended to dynamic systems is now derived. We decompose the kth estimates as

$$\bar{G}_k = P_k L_k U_k \tag{5.19}$$

where P_k is a permutation matrix, L_k is a lower triangular matrix, and U_k is an upper triangular matrix.

Then

$$\begin{aligned} Y_{k+1} &= P_k^T \\ X_{k+1} &= U_k^{-1} \end{aligned} \tag{5.20}$$

will triangularize the process.

This method is just to design a one-way decoupler iteratively. The input transformation X_k given as an analytic rational function of elements of $\bar{G}_k$. Hence it can be easily extended to the dynamical systems.

We show that X_k and Y_k of Equation 5.20 satisfy the assumptions of Lemma 1 in [48]. Let the output transformation P_k be identity without loss of generality. Let Θ be triangular, $Y = I$ and $X = I + \epsilon^k A + O(\epsilon^{k+l})$. Since X is an upper triangular matrix, A is an upper triangular one and has $n(n+1)/2$ arbitrary elements. Hence, with appropriate A having $n(n+1)/2$ arbitrary elements, we can have a triangular ΘA in

$$\begin{aligned} Y(\Theta + \epsilon^k \tilde{\Theta})X &= (\Theta + \epsilon^k \tilde{\Theta})(I + \epsilon^k A + O(\epsilon^{k+1})) \\ &= \Theta + \epsilon^k \Theta A + O(\epsilon^{k+1}) \end{aligned}$$

Applying this procedure to higher degrees of ϵ, we can obtain

$$Y(\Theta + \epsilon^k \tilde{\Theta})X \in \Omega.$$

This with Lemma 1 in [48] proves the convergence of this iterative method.

5.4 Simulation Examples

Two ill-conditioned systems from the literature are studied in the following two subsections(5.4.1, 5.4.2).

Table 5.1: Various uncertainty condition numbers for three distillation column processes

Process	$k(G)$	$k_R^*(G)$	$\gamma_\infty(G)$	$k^*(G)$	$\|\|\Lambda(G)\|\|_1$
G_{DV}	70.8	1.11	1.10	1.00	2.00
G_{LV}	142	142	86.4	138	138
G_{OR}	3671	223	206	7.37	7.75
G_{RTP}	181.7		90.6		169

5.4.1 Example 1

Consider the following ill-conditioned processes (Skogestad and Morari [85]):

$$G_{DV} = \begin{bmatrix} -0.878 & 0.014 \\ -1.082 & -0.014 \end{bmatrix}$$

$$G_{LV} = \begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix}$$

These processes appear in dual composition control of high-purity distillation. Various uncertainty condition numbers are shown in Table 5.1. Although G_{DV} is an ill-conditioned process, it is not sensitive to the element uncertainties, and classical element-by-element SISO identification will be sufficient. On the other hand, G_{LV} is very sensitive to the element uncertainties and iterative identification should be used. Table 5.2 shows this for a specific element uncertainty.

Iterative identification methods were applied to the process G_{LV} and identification steps were simulated by introducing artificial errors. MATLAB routines were used for the SVD, QR. and LU decompositions. Four iterative identification methods are compared in Table 5.3 in terms of $||\bar{G}G^{-l} - I||_\Omega, .2, \max|\sigma_i(\bar{G})/\sigma_j(G) - 1|$, and $||\bar{V}^T V - I||_\Omega$. Unless the identification error ϵ is the first two terms cannot be zero. But they can be considerably reduced with iterative identifications. For a

20% error in each identification step, all the methods converge quickly. The SVD method and the KM method provide better final estimates. As the identification error increases to 50%, the QR method and the KM method become worse.

Figures 5.3 and 5.4 show convergence aspects for two constant error matrices. All the methods give fast convergence for a small identification error of 10%, The best final estimates are obtained with the SVD method. Accuracy of the other methods is dependent on the identification error patterns. For the identification error pattern in Figure 5.3, the LU method provides the second-best result with respect to the accuracy of final estimates. On the other hand the KM method is the second-best for the identification error pattern in Figure 5.4. As the identification error increases, convergence becomes worse and even oscillatory. The SVD method and the LU method are best for this case in terms of convergence and accuracy of final estimates.

5.4.2 Example 2

Consider the following 3x3 distillation process (Ogunnaike and Ray [64]);

$$G_{OR} = \begin{pmatrix} 0.66 & -0.61 & -0.0049 \\ 1.11 & -2.36 & -0.012 \\ -34.68 & 46.2 & 0.87 \end{pmatrix}$$

Condition numbers for this process are shown in Table 5.1. The uncertainty condition number for performance is large, indicating high sensitivity of the element uncertainties, whereas a norm of RGA and the optimally scaled condition number are small. This means low influence of the element uncertainties on stability, as indicated in Table 5.2. To obtain models with robust performance, iterative identification methods can be applied. Figure 5.5 shows convergence aspects for a certain

Table 5.2: Effects of model uncertainties

Process	$\epsilon\Delta$	$\|\|\bar{G}G^{-1} - I\|\|_\Omega$	$\min_D \|\|D^{-1}ED\|\|_\Omega$	$\max_i \|[\sigma_i(G)/\sigma_j(G)] - 1\|$	$\|\|V^TV - I\|\|_\Omega$
G_{DV}	$0.2\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$	0.22	0.20	0.02	0.00
G_{LV}	$0.2\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$	28.3	27.1	27.6	0.046
G_{QR}	$0.2\begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}$	22.1	0.98	1.04	0.008
G_{RTP}	$02\begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$	23.5	18.4	0.897	0.529

Table 5.3: Iteration results for the G_{LV} process

ϵ	method	iteration 1 $\Delta = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$	iteration 2 $\Delta = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	iteration 3 $\Delta = \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix}$	iteration 4 $\Delta = \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix}$
0.2	SVD	(28.3, 27.6, 0.05)[a]	(2.60, 0.27, 0.02)	(0.21, 0.21, 0.00)	(0.20, 0.20, 0.00)
	KM		(2.60, 2.44, 0.01)	(0.24, 0.23, 0.00)	(0.29, 0.20, 0.00)
	QR		(11.2, 9.19, 0.00)	(2.56, 0.29, 0.02)	(0.63, 0.56, 0.00)
	LU		(9.78, 7.16, 0.02)	(1.77, 0.56, 0.01)	(0.42, 0.20, 0.00)
0.5	SVD	(70.9, 68.6, 0.14)	(20.0, 3.54 0.25)	(4.18, 4.14, 0.01)	(0.52, 0.52, 0.00)
	KM		(20.0, 16.4, 0.07)	(7.32, 2.83, 0.08)	(8.80, 5.11, 0.10)
	QR		(57.0, 54.8, 0.04)	(30.7, 14.8, 0.32)	(33.6, 20.1, 0.31)
	LU		(47.7, 42.0, 0.13)	(19.4, 10.9, 0.17)	(1.05, 0.52,0.01)

a: $(||\bar{G}G^{-l} - I||_\Omega, \max|[\sigma_i(\bar{G})\sigma_j(G)] - 1|.||\bar{V}V^T - I||_\Omega)$

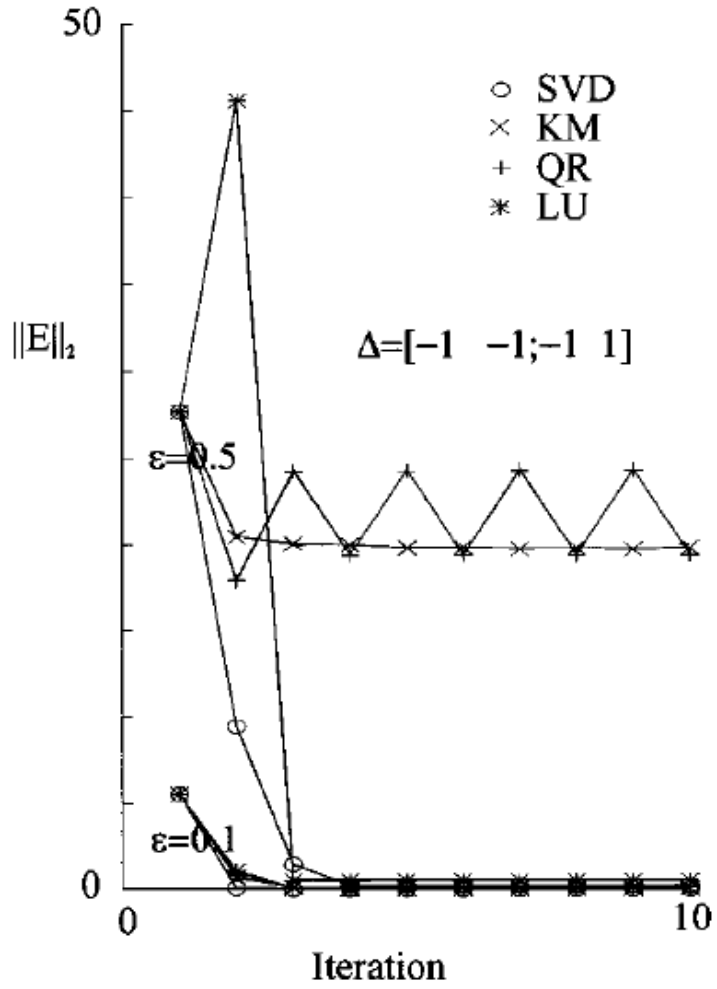


Figure 5.3: Convergence of iterative identification methods for the G_{LV} column with an identification error matrix Δ = [-1 -1; -1 1].

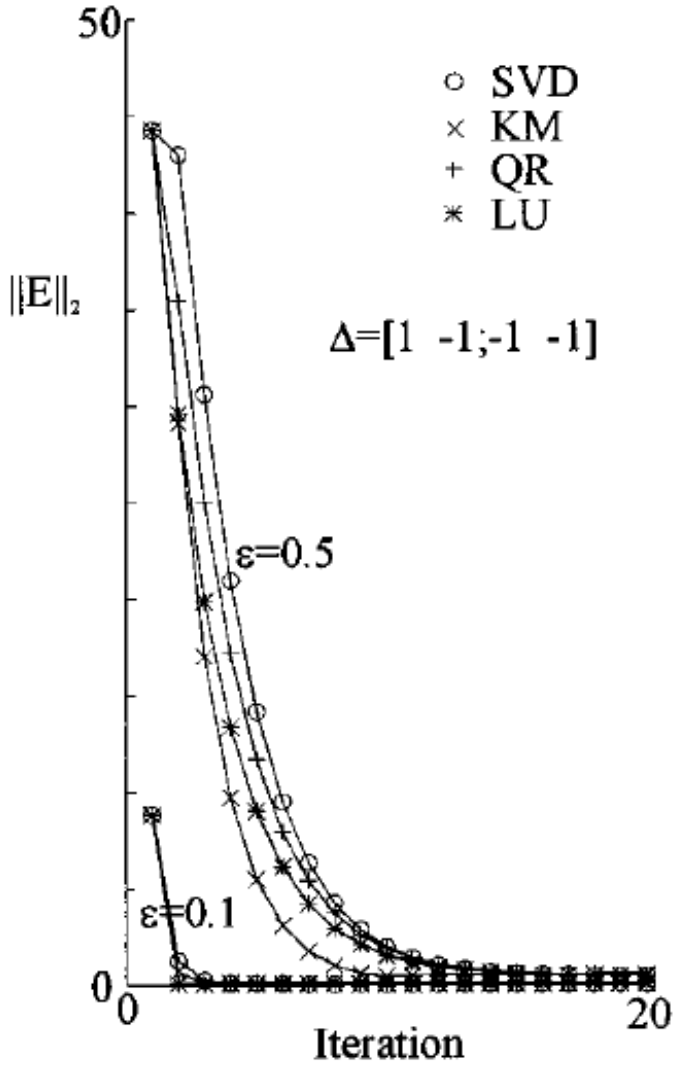


Figure 5.4: Convergence of iterative identification methods for the G_{LV} column with an identification error matrix Δ = [1 -1; -1 -1].

identification error pattern. The LU method is a little better for this process with an identification error of 50%.

5.4.3 RTP Example

The proposed iterative identification method is applied to identify the steady state gain matrix of RTP system around T=700°C whose nominal values are given as

$$G_{RTP} = \begin{pmatrix} 0.4518 & 0.5011 & 0.0505 \\ 0.4208 & 0.4590 & 0.0663 \\ 0.3823 & 0.4346 & 0.1036 \end{pmatrix}$$

Condition numbers for this RTP process are given in Table 5.1. All condition numbers are high as in G_{LV} process. This means that the RTP process is very sensitive to the element uncertainties. Table 5.2 shows this sensitivity. Iterative identification method is applied and Figure 5.6 shows convergence aspects for different identification error patterns.

5.5 Conclusions

Iterative identification methods are proposed for ill-conditioned processes that are very sensitive for the element uncertainties. For such processes, element uncertainties can be amplified and classical element-by-element identification cannot be used. Methods based on SVD, QR, and LU decompositions that diagonalize and triangularize the process iteratively with models identified incompletely are investigated. The convergence rate is dependent on the magnitude of identification error and error pattern. Simulations have shown that the SVD method has the largest convergence ranges of identification errors and other methods, including the

Koung and MacGregor method [45], have similar convergence ranges. The SVD, QR, and KM methods are all based on transformations with orthogonal matrices and hence can not easily be extended to obtain dynamic process transfer functions. On the other hand, the LU method uses analytical transformations and can be applied to dynamic processes.

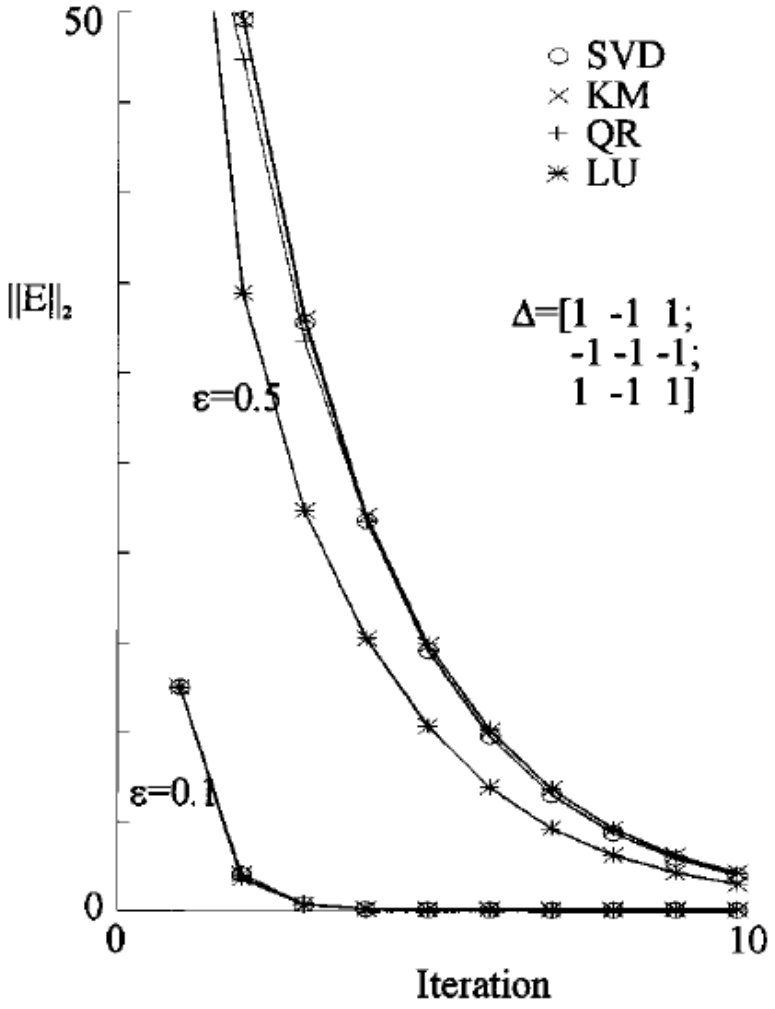


Figure 5.5: Convergence of iterative identification methods for the G_{OR} column with an identification error matrix Δ = [1 -1 1; -1 -1 -1; 1 -1 1].

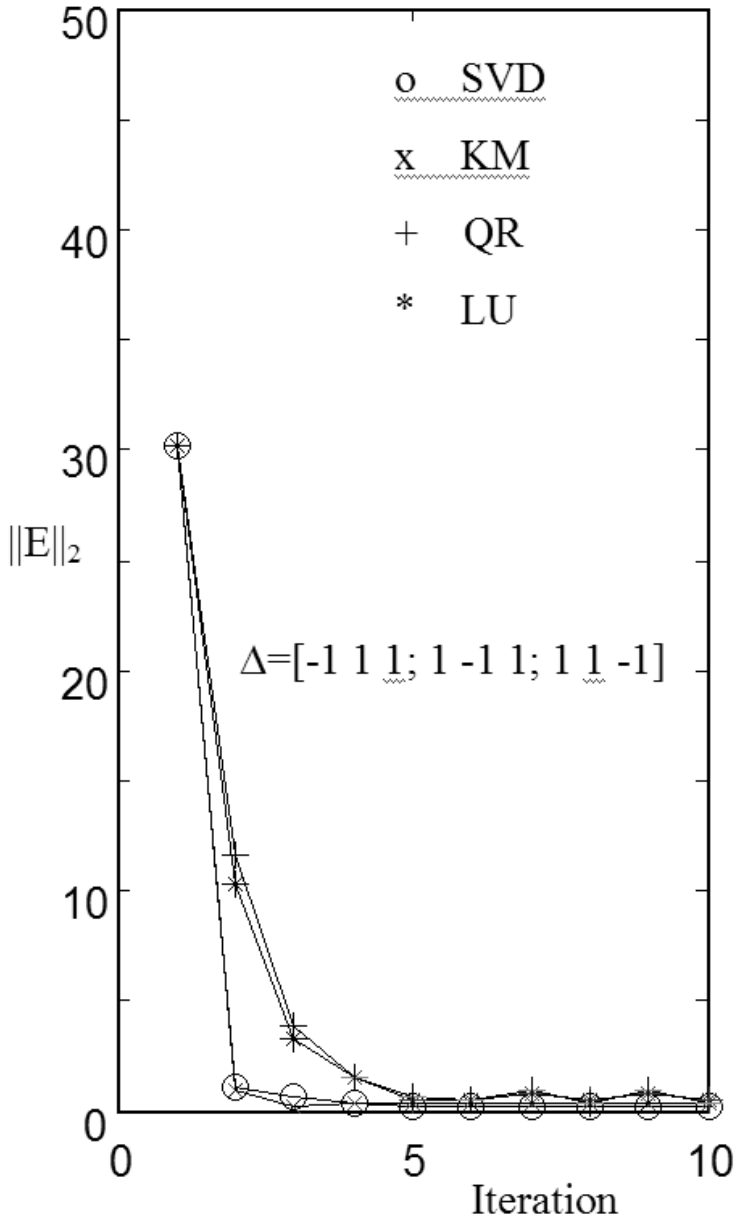


Figure 5.6: Convergence of iterative identification methods for the G_{RTP} column with an identification error matrix Δ = [-1 1 1; 1 -1 1; 1 1 -1].

Chapter 6

Effects of Diagonal input Uncertainties and Element Uncertainties in Control of Ill-Conditioned Processes

For some practical disturbances and uncertainties, acceptable control systems can be designed even for MIMO ill-conditioned processes. A class of ill-conditioned processes has been studied where an inversion based control system is not sensitive to the diagonal input uncertainties. We show for this class that element uncertainties arising from usual element-by-element model identification do not significantly degrade the control performance. Some useful measures for uncertainty effects on the control performance, stability, and inversion are proposed.

6.1 Introduction

Ill-conditioned processes are generally difficult to control (Skogestad et al., [88]), especially when full unstructured uncertainties and all possible disturbances are considered. However, for some ill-conditioned processes, control systems with acceptable performance and robustness can be designed. Two such cases were identified by Skogestad and Morari [84],[85]:

1. All major disturbances are aligned with the direction of the largest process gain so that a small diagonal controller can reject them sufficiently. The disturbance condition number can be used to determine whether the disturbances

are aligned with the direction of the largest process gain.

2. The uncertainties are only in the manipulated variables. There are some ill-conditioned processes for which the diagonal input uncertainties do not degrade control performances very much. Because uncertainties in the manipulated variables occur frequently in chemical processes, processes with high sensitivities to these uncertainties should be avoided.

The relative gain array (RGA) was used by Skogestad and Morari [85] to classify processes having problems with the diagonal input uncertainties. Large absolute RGA elements imply a large sensitivity for the diagonal input uncertainties. But small absolute RGA elements do not guarantee low sensitivity because they provide a rough lower bound of the uncertainty effects on control performances. A direct measure for the input diagonal uncertainties has been proposed by Chen et a1. [13]. They showed that the right minimally scaled condition number is a better measure, but it requires intensive singular value computations (Braatz and Morari [8]). Its lower and upper bounds were given by Chen and Nett [14] for simple estimation of the minimally scaled condition numbers.

Here a more efficient method is proposed to calculate measures for the sensitivity of uncertainties on the control performances, which we call the uncertainty condition number (UCN). Like the RGA of Skogestad and Morari [85], it can be very easily calculated. In addition, this method allows non diagonal uncertainties and indicates which input uncertainty is important to maintain the control performances. For some ill-conditioned processes, accurate model identification is cru-

cial to obtain good control performances [52]. Small element uncertainties arising from usual element-by-element identification will not guarantee good control performances. Here, measures for sensitivity of element uncertainties are given. Although the element uncertainties are full matrix, its measures are very similar to those of the diagonal input uncertainties. A system that is sensitive to the diagonal input uncertainties is also sensitive to the element uncertainties and vice versa. While input uncertainties are common in chemical processes, elementwise model uncertainties also occur frequently. Achieving low sensitivity to the element uncertainties is very important to obtain good control performances for ill-conditioned processes. Our result would enhance the possibility that inversion-based control systems can be safely used for processes with low right-minimally scaled condition numbers.

6.2 Diagonal Input Uncertainties

6.2.1 Performance

Uncertainties in the manipulated input variables and the sensor output variables always exist in chemical processes. It can be shown that small uncertainties in the sensor output variables do not affect control performances much if performances are measured in terms of process outputs. On the other hand, uncertainties in the manipulated input variables affect the control performances significantly for some ill-conditioned processes. Skogestad and Morari [85] identified a class of ill-conditioned processes, for which diagonal input uncertainties (uncertainties in the manipulated input variables) affect the control performances so much that tight control (e.g., inversion-based control) is very difficult. They used the norm of RGA as a measure

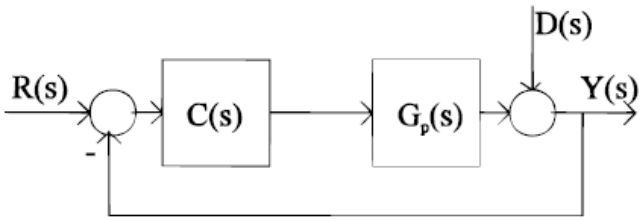


Figure 6.1: Feedback control system

of control difficulty. Later refinement by Chen et a1. [13] requires computation of structured singular values. In this paper a simpler measure is derived.

One performance measure of the control system is the ability to reject disturbances on the output, as shown in Figure 6.1. The closed-loop transfer function matrix between the output $Y(s)$ and the disturbance $D(s)$ is given as

$$Y(s) = (I + G_p(s)C(S))^{-1}D(s) \tag{6.1}$$

where $G_p(s)$ is an nxn process transfer function matrix containing uncertainties and $C(s)$ is the controller transfer function. For diagonal input uncertainties,

$$G_p(s) = G(s)(I + \Delta_D(s)) \tag{6.2}$$

The above closed-loop transfer function matrix 6.1 becomes

$$
\begin{aligned}
Y(s) &= [I + G(s)(I + \Delta_D(s))C(S)]^{-1}D(s) \qquad (6.3)\\
&\approx [I + G(s)C(s)]^{-1}[I - G(s)\Delta_D(s)G^{-1}(s)H(s)]D(s)\\
&\approx [I - H(s)C^{-1}(s)\Delta_D(s)C(s)][I + G(s)C(s)]^{-1}D(s)
\end{aligned}
$$

where $\Delta_D(s)$ is a diagonal uncertainty matrix and $H(s) = G(s)C(s)[I+G(s)C(S)]^{-1}$. For low-frequency regions where the complementary sensitivity function matrix $H(s)$ is near the identity matrix, the term

$$T_1(s) = G(s)\Delta_D(s)G^{-1}(s) \qquad (6.4)$$

or

$$T_2(s) = C^{-1}(s)\Delta_D(s)C(s) \qquad (6.5)$$

determines effects of the diagonal input uncertainties on disturbance rejection performance. Hereafter, we omit the Laplace variable s because only the steady state gain matrix is considered. Refer to Appendix B for notation used in subsequent equations.

Instead of exploiting quantity 6.4 or 6.5 directly, Skogestad and Morari [85] used the RGA,

$$\Lambda(G) = \lambda_{ij} = G^{-T} \circ G \qquad (6.6)$$

where $\circ$ denotes the Hadamard product. For diagonal input uncertainties, the RGA and T_1 in equation 6.4 are related as [63]

$$(G\Delta_D G^{-1})_{jj} = \sum_{j=1}^{n} \lambda_{ij}\Delta_{jj} \qquad (6.7)$$

Large RGA elements become large diagonal elements in $G\Delta_D G^{-1}$, leading to a large sensitivity for the diagonal input uncertainties. However, the converse is not true. That is, small RGA elements do not guarantee that the magnitude of $G\Delta_D G^{-1}$ is small, because off-diagonal elements of $G\Delta_D G^{-1}$ can be large.

To find a more direct measure of diagonal input uncertainty effects, Chen et ai. [13] proposed a worst case deviation,

$$\eta_{CFN}(G) \equiv \max_{||\Delta_D||_\Omega=1} ||G\Lambda_D G^{-1}||_\Omega \tag{6.8}$$

They showed that

$$\eta_{CFN}(G) = \mu^2 \left(\begin{bmatrix} 0 & G^{-1} \\ G & 0 \end{bmatrix} \right) \tag{6.9}$$

with the same uncertainty structure as in the right minimally scaled condition number $k_R(G)$ (Braatz and Morari [8], Chen et al.[13]) and

$$||\Lambda(G)||_{j\infty} \leq \eta_{CFN}(G) \leq k_R(G) \tag{6.10}$$

The upper bound in 6.10 is known to be very tight. Unlike the RGA, there is no ambiguity in estimating the diagonal input uncertainty effects. However, the computation of the structured singular value or its upper bound is complex.

Here we propose a computationally simpler measure called the uncertainty condition number (UCN) defined as

$$\eta_p(G) \equiv \max_{||\Delta_D||_p=1} ||G\Delta_D G^{-1}||_p \tag{6.11}$$

Specifically

$$\eta_2(G) \equiv \max_{||\Delta_D||_F=1} ||G\Delta_D G^{-1}||_F \eta_\infty(G) \equiv \max_{||\Delta_{ii}||\leq 1} \{\max_{ij}\{|(G\Delta_D G^{-1})_{ij}|\}\}$$

Although the matrix norm $||\cdot||_p$ is not as popular as the induced matrix norm $||\cdot||_{ip}$, it (especially the Frobenius norm) can be often adopted to represent the magnitude of the matrix.

Then we have

$$\eta_p(G) \equiv \max_{||vec(\Delta_D)||_p=1} ||\mathrm{vec}(G\Delta_D G^{-1})||_p \tag{6.12}$$

and

$$\begin{aligned} \mathrm{vec}(\mathrm{G_D G^{-1}}) &= (G^{-T} \otimes G)\mathrm{vec}(\Delta_\mathrm{D}) \\ &= \Phi \begin{bmatrix} \Delta_{11} \\ \Delta_{22} \\ \vdots \\ \Delta_m \end{bmatrix} \end{aligned} \tag{6.13}$$

where $\otimes$ denotes the Kronecker product. The n^2xn matrix Φ is formed by removing columns of $G^{-T} \otimes G$ that correspond to zero elements of $vec(\Delta_D)$, giving the following result.

$$\eta_p(G) = ||\Phi||_{ip} \tag{6.14}$$

Proof: This is self-evident from equation 6.13 and the definition of the in-

duced matrix norm of eq A8. The Φ matrix contains the RGA. That is,

$$\Lambda = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,n} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ \phi_{k,1} & \phi_{k,2} & \cdots & \phi_{k,n} \\ \phi_{k+1,1} & \phi_{k+1,2} & \cdots & \phi_{k+1,n} \\ \vdots & \vdots & \cdots & \vdots \\ \phi_{n^2,1} & \phi_{n^2,2} & \cdots & \phi_{n^2,n} \end{bmatrix} \tag{6.15}$$

The following relations can be verified (Horn and Johnson [38], Golub and Van Loan [31];

$$\begin{aligned} ||\Lambda||_\infty \leq \eta_\infty(G) &\leq \sqrt{n}\eta_2(G) \leq n\sqrt{n}\eta_\infty(G) \\ \frac{1}{\sqrt{n}}\eta_2(G) &\leq \eta_{\mathrm{CFN}}(G) \leq \sqrt{n}\eta_2(G) \\ \eta_\infty(G) &\leq \eta_{\mathrm{CFN}}(G) \leq n\eta_\infty(G) \end{aligned} \tag{6.16}$$

These show that either $\eta_{\mathrm{CFN}}(G), \eta_2(G), \mathrm{or}\eta_\infty(G)$ can be used without incorrectly estimating the diagonal input uncertainty effects.

6.2.2 Remarks

With the above uncertainty condition numbers, we can extract some useful observations that are not obvious from the RGA analysis of Skogestad and Morari [85]. Uncertainty condition numbers are invariant for input scaling. That is,

$$\eta_p(G) = \eta_p(GD_2) \qquad D_2 = \text{a diagonal matrix} \tag{6.17}$$

Hence, if $\eta_p(G)$ is large, $\eta_p(GC)$ is always large for a diagonal controller C and consequently GC is ill-conditioned. A diagonal controller cannot correct directionality when $\eta_p(G)$ is large. On the other hand, if $\eta_p(G)$ is small, $k_R(G) \approx \eta_{\mathrm{CFN}}$ is small and

the directionality of G can be corrected by a diagonal controller. For the extreme case when the right singular vector matrix of G is identity, $\eta_2(G) = k_R(G) = 1$ and perfect correction of the directionality of G can be obtained with a diagonal controller. As shown by Skogestad and Morari [85], small $||\Lambda||$ (e.g., $\Lambda = I$) does not guarantee that the directionality of G can be corrected by a diagonal controller.

If $\eta_p(G)$ is large, we cannot use an inversion-based (decoupling) control system due to its sensitivity to the diagonal input uncertainties. Many people believe that a one-way decoupling control system has a lower sensitivity, but this is not always true. Here we show that any attempt to correct the directionality of illconditioned processes causes a sensitivity problem for the diagonal input uncertainties. Since

$$\begin{aligned} \mathrm{vec}(G\Lambda_D G^{-1}) &= \mathrm{vec}((\mathrm{GC})\mathrm{C}^{-1}\Lambda_\mathrm{D}\mathrm{C}(\mathrm{GC})^{-1}) \\ &= ((GC)^{-T} \otimes (GC))\mathrm{vec}(\mathrm{C}^{-1}\Lambda_\mathrm{D}\mathrm{C}) \end{aligned} \tag{6.18}$$

we have

$$||\mathrm{vec}(\mathrm{G}\Lambda_\mathrm{D}\mathrm{G}^{-1})||_\mathrm{p} \leq ((\mathrm{GC})^{-\mathrm{T}} \otimes (\mathrm{GC})||_\mathrm{ip}||\mathrm{vec}(\mathrm{C}^{-1}\Lambda_\mathrm{D}\mathrm{C})||_\mathrm{p} \tag{6.19}$$

This equation shows that, if $\eta_p(G)$ is large and GC is well-conditioned, then $\eta_p(C^{-1})$ is large. That is, when $\eta_p(G)$ is large, any control system that corrects the directionality of ill-conditioned processes will suffer from a sensitivity problem due to the diagonal input uncertainties. A one-way decoupler is not an exception, as illustrated by a later example.

A one-way decoupler can be utilized for reasons other than correcting the directionality. When $\eta_p(G)$ is large, a large diagonal controller gain can be used

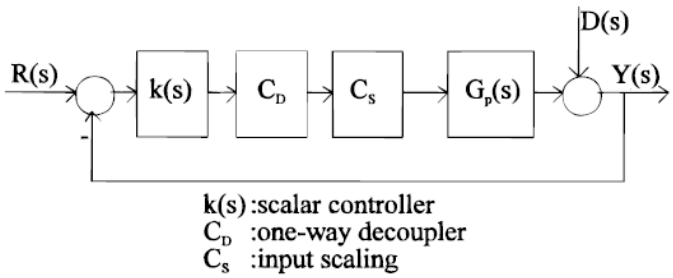


Figure 6.2: Proposed control system structure for processes with large uncertainty condition numbers.

to obtain desired control performances while not increasing the sensitivity to the diagonal input uncertainties. For some processes, a large diagonal controller gain cannot be applied due to interactions, and a one-way decoupler can be utilized to decrease interactions. We can always design a small gain one-way decoupler whose elements are all less than or equal to 1 (see Chapter 4 in Golub and Van Loan [31]). It will reduce interactions without changing the directionality very much, enabling a large gain scalar controller to be applied. For processes with large $\eta_p(G)$, we may use the control systems in Figure 6.2, where the input scaling is such that the right minimal condition number is obtained.

In the controller structure selection guide for the ill-conditioned processes of Skogestad and Morari [85], the RGA criterion can be replaced by the uncertainty condition number $\eta_p(G)$. Particularly, $\eta_\infty(G)$ is useful because it is computationally simple and would be sufficient to determine the effect of uncertainty on the control performances.

The Φ matrix also contains information about which input uncertainty has more impact on the control performance. For this purpose, the column absolute sums, $\sum_{j=1}^{n} |\phi_{ij}|$, can be used. The larger the sum, the more impact the corresponding input uncertainty has on the control performances. In the analysis, nondiagonal uncertainties can also be considered easily by keeping the corresponding columns in equation 6.13.

6.3 Element Uncertainties

6.3.1 Performance

MIMO models are often obtained by perturbing inputs one at a time. This usually provides sufficient accuracy for determining transfer function elements. However, for some ill-conditioned processes, small errors in the individual gain elements can cause large multiplicative errors in the overall open-loop transfer function matrix, leading to poor closed-loop performances. For such processes, a different identification scheme as in Koung and MacGregor [44][45], Li and Lee [52]. and Lee et al. [48] should be used. But not all ill-conditioned processes suffer from this difficulty, as shown below.

Element uncertainties are modeled as

$$G_p = G + G \circ \Delta_G \tag{6.20}$$

where Δ_G is a full uncertainty matrix. Then

$$G_p C = (I + (G \circ \Delta_G) G^{-1}) GC \tag{6.21}$$

The quantity $(G \circ \Delta_G)G^{-1}$ plays the same role as $G\Delta_D G^{-l}$ in the case of the diagonal input uncertainties.

Expressions for the following uncertainty condition numbers can be developed:

$$\gamma_p(G) \equiv \max_{||\Delta_G||_p=1} ||(G \circ \Delta_G)G^{-l}||_p \tag{6.22}$$

As before, we have

$$\begin{aligned} \mathrm{vec}((\mathrm{G} \circ \Delta_\mathrm{G})\mathrm{G}^{-1}) &= (G^{-T} \otimes I)\mathrm{vec}(\mathrm{G} \circ \Delta_\mathrm{G}) \\ &= (G^{-T} \otimes I)\mathrm{diag}(\mathrm{vec}(\mathrm{G}))\mathrm{vec}(\Delta_\mathrm{G}) \\ &= \Psi \mathrm{vec}(\Delta_\mathrm{G}) \end{aligned} \tag{6.23}$$

From equation 6.23, the following result can be obtained.

$$\gamma_p(G) = ||\Psi||_{ip} \tag{6.24}$$

From

$$
\begin{aligned}
(G \circ \Delta_G)G^{-1} \quad = \quad & \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \times \\
& \begin{bmatrix} [\Delta_G]_{11} & 0 & \cdots & 0 \\ 0 & [\Delta_G]_{12} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & [\Delta_G]_{1n} \end{bmatrix} G^{-1} + \\
& \begin{bmatrix} 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \times \\
& \begin{bmatrix} [\Delta_G]_{21} & 0 & \cdots & 0 \\ 0 & [\Delta_G]_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & [\Delta_G]_{2n} \end{bmatrix} G^{-1} + \ldots
\end{aligned} \tag{6.25}
$$

and equations B.20 and 6.13, it can be verified that

$$\eta_\infty = \gamma_\infty(G) \tag{6.26}$$

Hence processes with a large (small) sensitivity for the diagonal input uncertainties will have a large (small) sensitivity for the element uncertainties. From the column absolute sums of Ψ the most important gain element which should be estimated accurately and kept invariant can be also found.

6.3.2 Inversion

The amount of changes in the inverse of the gain matrix due to element changes is further evidence of sensitivity. Let the relative element error in inverse

be defined as

$$G_p^{-l} = G^{-1} + G^{-1} \circ \Delta_{G^{-l}} \tag{6.27}$$

It is well-known (Grosdidier et al. [35]) that $d[\Delta_{G^{-l}}]_{ji} = -\lambda_{ij} d[\Delta_G]_{ij}$. This equation indicates that processes with large RGA elements are sensitive to modeling error (Skogestad and Morari [85]. Effects on the other elements of the inverse are derived here.

For sufficiently small $||\Delta_G||$,

$$\begin{aligned} G_p^{-l} &= (G + G \circ \Delta_G)^{-1} \\ &= G^{-1}(I + (G \circ \Delta_G)G^{-1})^{-1} \\ &\approx G^{-1} - G^{-1}(G \circ \Delta_G)G^{-1} \end{aligned} \tag{6.28}$$

Equating 6.27 and 6.28 and applying the vec$(\cdot)$ operation, we have

$$\begin{aligned} \mathrm{vec}(\Delta_{\mathrm{G}^{-1}}) &\approx (\mathrm{diag}(\mathrm{vec}(\mathrm{G}^{-1})))^{-1}(\mathrm{G}^{-\mathrm{T}} \otimes \mathrm{G}^{-1}) \times \\ &\quad (\mathrm{diag}(\mathrm{vec}(\mathrm{G})))\mathrm{vec}(\Delta_{\mathrm{G}}) \\ &\equiv \Theta \mathrm{vec}(\Delta_{\mathrm{G}}) \end{aligned} \tag{6.29}$$

The $n^2 \times n^2$ matrix Θ has full information about incremental effects of the element uncertainties on the inverse. That is,

$$d\mathrm{vec}(\Delta_{\mathrm{G}^{-1}}) = \Theta \mathrm{dvec}(\Delta_{\mathrm{G}}) \tag{6.30}$$

The following uncertainty condition number can be defined:

$$\zeta_p \equiv \max_{||d\Delta_G||_p = l} ||d\Delta_G^{-1}||_p \tag{6.31}$$

$$\zeta_p = ||\Theta||_{ip} \tag{6.32}$$

When the element uncertainty becomes large, the process gain matrix G_p can be singular. The singularity of G_p is related to the right-half-plane zeros, which are usually detrimental for control. It is well-known that, if a single element in G_p is perturbed, a relative change $-l/\lambda_{ij}$ in its corresponding element causes G_p to be singular (Yu and Luyben [106], Hovd and Skogestad [39]). For simultaneous perturbations, we can have several results. For one-row perturbations, that is, $(G_p)_{ij} = G_{ij}(1 + \Delta_G)_{ij}, j = 1, 2, \ldots, n$, the matrix G_p becomes singular if and only if

$$1 + \sum_{j=1}^{n} (\Delta_G)_{ij} \lambda_{ij} = 0 \tag{6.33}$$

Proof of equation 6.33 is given in Lee et al.[47]. A necessary and sufficient norm condition for the non singularity of G_p can be obtained with the Hölder inequality as

$$\max_j(|(\Delta_G)_{ij}|) < \frac{1}{\sum_{j=1}^{n} |\lambda_{ij}|} \tag{6.34}$$

A similar result can be obtained for one-column perturbations.

If uncertainties are only contained in the diagonal elements, then

$$\begin{aligned} G_p = G + G \circ \Delta_D \quad &= [I + (G \circ \Delta_D) G^{-1}] G \\ &= [I + \Delta_D \tilde{G} G^{-1}] G \end{aligned} \tag{6.35}$$

where $\tilde{G} = \text{diag(G)}$. Hence, a necessary and sufficient condition for the non singularity of G_p is

$$\max_i(|(\Delta_G)_{ii}|) < 1/\mu(\tilde{G}G^{-1}) \tag{6.36}$$

The matrix $\tilde{G}G^{-1}$ is known as the performance relative gain array (PRGA). It appears in Stanley et al.[90], Grosdidier[33], and Hovd and Skogestad[39] for the analysis of sensitivity of multiloop control systems.

For the full element uncertainties, some sufficient conditions for the nonsingularity of G_p as

$$||\Delta_G||_{i2} < \frac{1}{k^*(G)} \tag{6.37}$$

can be obtained from the nonsingularity condition

$$||(G \circ \Delta_G)G^{-1}||_{i2} \leq ||G||_{i2}||\Delta_G||_{i2}||G^{-1}||_{i2} < 1$$

The nonsingularity bound on Δ_{G_∞} is between $1/(nk^*(G))$ and $1/\max(\Lambda_i l, \Lambda_{i\infty})$.

6.3.3 Remarks

Because $(G \circ \Delta_G)G^{-1} = (GD \circ \Delta_G)(GD)^{-1}$, $\gamma_P(G) = \gamma_P(GD)$ for a positive diagonal D. Hence it will be similar with $\eta_p(G)$ and $k_R(G)$ which are both invariant for the input scaling. On the other hand, $\zeta_p(G)$ is invariant for the input and output scalings. It will have a similar value as RGA and $k^*(G)$. When right singular vectors are known *a priori*, Koung and MacGregor [44] proposed a MIMO model identification method for ill-conditioned processes. Since G_pV is identified element-by-element, the modeling uncertainties become

$$G_pV = GV + (GV) \circ \Delta \tag{6.38}$$

where V is the right singular vectors matrix. We have

$$\begin{aligned} G_p &= G + ((GV) \circ \Delta)V^T \\ &= [I + ((GV) \circ \Delta)V^T G^{-1}]G \\ &= [I + ((U\Sigma) \circ \Delta)\Sigma^{-1}U^T]G \\ &= [I + (U \circ \Delta)U^T]G \end{aligned} \tag{6.39}$$

where (U, Σ, V) is the singular value decomposition of G. Since $\gamma_2(U) = 1$, there is no amplification of the model identification uncertainties. If the right singular vectors are known, a model appropriate for control system design can be obtained with an input transformation, as shown by Koung and MacGregor[44].

Li and Lee[52] identified a process model by fitting both the process gain matrix and its inverse obtained by finding RGA experimentally. When ζ_P is small, their method cannot be used because both the process gain matrix and its inverse can have small relative error. Their method can only be used for processes with large ζ_P or large RGA elements.

6.4 Summary of Various Measures

Various measures for uncertainty effects have been proposed. These can be summarized as follows.

(i) Number that is variant to the input and output scaling:

Condition number: $k(G) = ||G||_{i2}/||G^{-1}||_{i2}$

(ii) Numbers that are invariant to the input scaling:

Right minimally scaled condition number,

$k_R(G) = \min k(GD)$

Effect on the performance of diagonal input uncertainties,

$\eta_p(G) = \max_{||\Delta_D||_P=1} ||G\Delta_D G^{-1}||_P$

$= ||\Phi||_{ip} \qquad \Phi_{ij} = (G^{-T} \otimes G)_{ij(n+1)-n}$

Effect on the performance of element uncertainties,

$\gamma_p(G) \equiv \max_{||\Delta_G||_P=1} ||G\Delta_D G^{-1}||_P$

$= ||\Psi||_{ip} \qquad \Psi = (G^{-T} \otimes I)\text{diag}(\text{vec(G)})$

(iii) Numbers that are invariant to the input and output scaling

Minimally scaled condition number, $k^*(G) = \min k(D_1 G D_2)$

Minimally scaled numbers in (ii), $\eta_P^*(G) = \min \eta_P(DG)$

$\gamma_P^*(G) = \min \gamma_P(DG)$ Norm of RGA, $||\Lambda||$, $\Lambda = G^{-T} \circ G$

Effect on the inversion of element uncertainties,

$$\zeta_P(G) \equiv \max_{||d\Delta_G||_P=1} ||d\Delta_{G^{-1}}||_P$$

$$= ||\Theta||_{ip}$$

where

$$\Theta = -(\text{diag}(\text{vec}(G^{-1})))^{-1} \times (G^{-T} \otimes G^{-1})(\text{diagvec}(G)))$$

6.5 Examples

There are three examples discussed; the first and second examples are test problems and the third example is an RTP system.

6.5.1 Example 1

Consider the 2 x 2 process gain matrix:

$$G = \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix}$$

The matrices Φ and Ψ can be calculated as

$$\begin{aligned} \text{vec}(G\Delta_D G^{-1}) &= (G^{-T} \otimes G)\text{vec}(\Delta_D) \\ &= \begin{bmatrix} \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} & -100 \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} \\ 0 \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta_{11} \\ 0 \\ 0 \\ \Delta_{22} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 100 & -100 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_{11} \\ \Delta_{22} \end{bmatrix} = \Phi \begin{bmatrix} \Delta_{11} \\ \Delta_{22} \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \text{vec}((G \circ \Delta_G)G^{-1}) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 100 & 0 & -100 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_{11} \\ \Delta_{21} \\ \Delta_{12} \\ \Delta_{22} \end{bmatrix} \\ &= \Psi \begin{bmatrix} \Delta_{11} \\ \Delta_{21} \\ \Delta_{12} \\ \Delta_{22} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} G\Delta_D G^{-1} &= \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} \begin{bmatrix} \Delta_{11} & 0 \\ 0 & \Delta_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -100 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \Delta_{11} & 0 \\ 100\Delta_{11} - 100\Delta_{22} & \Delta_{22} \end{bmatrix} \end{aligned}$$

and

$$
\begin{aligned}
(G \circ G)G^{-1} &= \left(\begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} \circ \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix}\right) \begin{bmatrix} 1 & 0 \\ -100 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \Delta_{11} & 0 \\ 100\Delta_{21} - 100\Delta_{22} & \Delta_{22} \end{bmatrix}
\end{aligned}
$$

Various uncertainty condition numbers are calculated in Table 6.1. It is obvious that there are considerable uncertainty effects on the control performances in inversion-based control systems, but as indicated by several investigators (Skogestad and Morari [85], Chen et al [13]), the RGA ($||\lambda||_1 = 2$) gives the wrong conclusion. The above uncertainty condition numbers ($\eta_\infty = \nu_\infty = 200$) show that achieving robustness in control system performance will be difficult for diagonal input uncertainties and/or element-wise model uncertainties.

Absolute column sums of Φ show that both input variables are equally important and those of Ψ show that better estimation of g_{21} and g_{22} is more important than the other gains for control performance robustness.

6.5.2 Example 2

Consider the high-purity distillation cases of Skogestad and Morari [84] [85];

$$
G_{LV} = \begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix}
$$

and

$$
G_{DV} = \begin{bmatrix} -0.878 & 0.014 \\ -1.082 & -0.014 \end{bmatrix}
$$

Uncertainty condition numbers are calculated in Table 6.1. All measures indicate high sensitivity in the LV column and low sensitivity in the DV column

Table 6.1: Various Measures for the Uncertainty Effects, Examples 1-3

	example 1	example 2 G_{LV}	example 2 G_{DV}	example 3
$k(G)$	10.002	141.7	70.8	124.6
$k_R(G)$	200 (200/100)^a	141.7 (141.7/70.87)^a	1.110 (2.011/1.005)^a	77.5 (82.0/20.5)^a
$\eta_2(G) = \|\|\Phi\|\|_{i2}$	141.4	100.2	1.011	40.8
$\eta_\infty(G) = \|\|\Phi\|\|_{j\infty}$	200	86.43	1.104	31.2
$\nu_2(G) = \|\|\Psi\|\|_{i2}$	141.4	78.27	0.797	23.8
$\nu_\infty = \|\|\Psi\|\|_{j\infty}$	200	86.43	1.104	31.2
$k^*(G)$	1.173	138.3	1	66.2
$\|\|\Lambda\|\|_1$	2	138.3	2	66.1
$\zeta_2(G)$	not applicable	138.3	1.110	216
$\zeta_\infty(G)$	not applicable	139.3	2.10	414

^{a} upper bound/lower bound by Chen and Nett[14]

for the diagonal input uncertainties and element uncertainties. It can be shown by their inverses when element uncertainties are added,

$$\begin{aligned}(G_{LV} + G_{LV} \circ \Delta)^{-1} &= \begin{bmatrix} 0.878(1+\Delta_{11}) & -0.864(1+\Delta_{12}) \\ 1.082(1+\Delta_{21} & -1.096(1+\Delta_{22}) \end{bmatrix}^{-1} \\ &\approx \frac{1}{d_1} \begin{bmatrix} 39.94 & -31.49 \\ 39.43 & -32.00 \end{bmatrix} \\ d_1 &\approx 1 + 3506(\Delta_1 1 + \Delta 22) - 34.07(\Delta_{12} + \delta_{21}) \end{aligned}$$

and

$$\begin{aligned}(G_{DV} + G_{DV} \circ \Delta)^{-1} &= \begin{bmatrix} -0.878(1+\Delta_{11}) & 0.014(1+\Delta_{12}) \\ -1.082(1+\Delta_{21} & -0.014(1+\Delta_{22} \end{bmatrix}^{-1} \\ &\approx \frac{1}{d_2} \begin{bmatrix} -0.510 & -0.510 \\ 39.43 & -32.00 \end{bmatrix} \\ d_2 &\approx 1 + 0.448(\Delta_1 1 + \Delta 22) + 0.552(\Delta_{12} + \delta_{21}) \end{aligned}$$

For the LV column, the inverse of the gain matrix can differ considerably from nominal values for small element-wise errors. Even its sign can be changed by a 1% change in some elements. Hence the usual element by element identification may not provide a satisfactory model for control system design.

Finally, we show that the one-way decoupler with directionality correction is also very sensitive to the input diagonal uncertainties. Consider the LV column. A small gain one-way decoupler can be designed as

$$C_D = \begin{bmatrix} 1 & 0 \\ 0.9872 & -1 \end{bmatrix}$$

Then we have

$$G_{LV} C_D = \begin{bmatrix} 0.0251 & 0.864 \\ 0 & 1.096 \end{bmatrix}$$

A one-way decoupler with directionality correction is designed as

$$C_1(s) = \begin{bmatrix} 1 & 0 \\ 0.9872 & -1 \end{bmatrix} \begin{bmatrix} 39.84 & 0 \\ 0 & 0.912 \end{bmatrix} \frac{1}{s}$$

It can be compared with a one-way decoupler without directionality correction, which is

$$C_2(s) = \begin{bmatrix} 1 & 0 \\ 0.9872 & -1 \end{bmatrix} \frac{25}{s}$$

Control performances for processes with and without 20% diagonal input uncertainties, i.e.,

$$\Delta_D = \begin{bmatrix} 0.2 & 0 \\ 0 & -0.2 \end{bmatrix}$$

are simulated in Figure 6.3. For the normal process without uncertainties, set point responses are well balanced for the one-way decoupler with the directionality correction. But we can see that its performance deteriorates very much when uncertainties are introduced. On the other hand, normal control performances for the one-way decoupler without the directionality correction are not symmetric but they are not affected very much by the diagonal input uncertainties.

6.5.3 Example 3

The rapid thermal processing (RTP) system is one of the key semiconductor manufacturing technologies, where a single wafer is exposed to high temperatures for short period of time. One obstacle in designing a control system to maintain temperature uniformity across the wafer is the ill-conditioned nature of some RTP systems (Schaper et al.[76]; Stuber et al.[91]). A typical process gain matrix of the

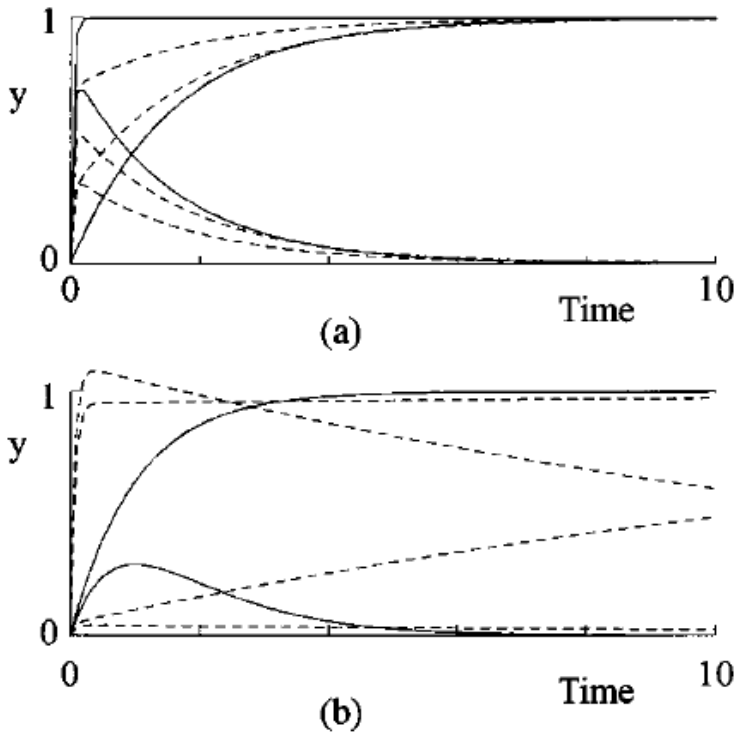


Figure 6.3: Set point responses for the LV column with one-way decoupling control systems: (a) without directionality correction; (b) with directionality correction; (solid line) nominal process; (dotted line) perturbed process with 20% input uncertainties.

Texas Instruments RTP system given by Schaper et al.[76] is

$$G = \begin{bmatrix} 4.8 & 5.4 & 4.4 & 1.0 \\ 4.5 & 5.9 & 4.7 & 1.1 \\ 2.9 & 5.5 & 6.5 & 1.4 \\ 1.8 & 3.7 & 7.3 & 2.1 \end{bmatrix}$$

The uncertainty condition numbers in Table 6.1 show that the inversion-based control system such as MIMO IMC controller used by Schaper et al.[76] will not easily achieve robust performance for the manipulated input uncertainties and elementwise model uncertainties.

6.6 Conclusions

We have presented new measures of some practical uncertainty effects on the closed-loop control performances, stability, and model inversion, which build on previous works by Skogestad and Morari[85] and Chen et al.[13]. The proposed simpler norm bounds (the uncertainty condition numbers) are induced norms of matrices that can be constructed very easily, like the RGA. In addition to the diagonal input uncertainties, this analysis permits non diagonal uncertainties and elementwise model uncertainties with minor modifications. Key observations with the uncertainty condition numbers follow.

(1) A diagonal controller can correct the directionality of ill-conditioned processes if the uncertainty condition number is small. The RGA indicates that a diagonal controller cannot correct the directionality if elements of the RGA are large.

(2) Any controller that corrects the directionality of ill-conditioned processes suf-

fers from sensitivity to diagonal input uncertainties if the uncertainty condition number is large. A one-way decoupler, which is believed to be insensitive to uncertainties, is not an exception.

(3) A process that has low sensitivity for diagonal input uncertainties also has low sensitivity for the element uncertainties, which come from element-by-element identification.

(4) Model identification methods for ill-conditioned processes such as presented by Koung and MacGregor[44] provide good models with uncertainties that do not degrade control performances significantly.

(5) Our analysis has provided information about which inputs and gain elements are important for control performance. The nonlinearity of sensitive inputs that cause input uncertainties should be kept low, and sensitive gain elements, should be identified accurately.

(6) Sensitivity equations when the process gain matrix is inverted are also obtained.

Chapter 7

Multiloop Controller Design

7.1 Introduction

Multiloop single-input single-output controllers are often used for controlling interacting multivariable processes because of their simplicity in implementation, namely, they are easily understandable to control engineers and require fewer parameters to tune than multivariable controllers. Another advantage of multiloop controllers is that loop failure tolerance of the resulting control system can be easily obtained. Since, some loops can be in manual mode or the manipulated variables of some loops can be saturated to their limits, the loop failure tolerance is important for practical applications.

Three types of tuning methods for multiloop control systems are available:

- Detuning method
- Sequential closing method (Mayne [59])
- Independent design method (Grosdidier and Morari [34] , Skogestad and Morari [87])

In the detuning method, each controller of the multiloop system is first designed ignoring process interactions from the other loops. Then interactions are

taken into account and each controller is detuned until some performance criterion is met. One of the methods of this type is the biggest log-modulus tuning (BLT) method proposed by Luyben [57]. Each controller of the multiloop control system is designed with the Ziegler-Nichols tuning for the paired transfer functions. Then detuning is performed by adjusting one parameter F that divides and multiplies the controller gain and the integral time, respectively. The detuning parameter F is increased from 1 until the biggest log-modulus, which is a measure of how far the control system is from being unstable, becomes a given value. The method is very simple to apply.

In the sequential closing method (Mayne [59]), each controller is designed sequentially. A controller for a selected input-output pair is designed and this loop is closed. Then a second controller is designed for a second pairing while the first controller is closed, and so on. Since each controller is designed using SISO methods, it is as simple as the detuning method. A potential disadvantage is that failure tolerance is not guaranteed automatically when the loops that are designed earlier fail. Also, the method depends strongly on which loop is designed first and how this controller is designed. Usually, the fast loop is designed first. Chiu and Arkun [16] discuss how to handle some properties such as robustness, but these modifications tend to be complex. The relay feedback autotuning method of Åström and Hägglund [4] can be used for each SISO controller tuning, as discussed by Shen and Yu [80] and Loh et al. [54]. When the pairing is wrong, this sequential autotuning is unsatisfactory and on-line experiments should be repeated with different pairings. Identification of all elements of the transfer function matrix and applying various

methods for pairing and tuning are preferable.

In the independent design method, each controller is designed based on the paired transfer functions while satisfying some constraints due to the process interactions (Grosdidier and Morari [34], Skogestad and Morari [87]). The constraints imposed on the individual design are given by criteria such as the μ-interaction measure of Grosdidier and Morari [34]. Usually, failure tolerance is automatically obtained in the independent design method. One disadvantage of the method is that it is conservative due to the assumption about controllers in the other loops (Skogestad and Morari [87]).

Most design examples in the independent design method are carried out by specifying the form of each closed-loop transfer function. This results in the IMC-PID-type controller settings having one tuning parameter for each loop. As the IMC-PID tuning parameter varies, only the gain of each controller varies while the integral time and the derivative time remain constant. A multiloop PI controller tuning method that utilizes this characteristics of the IMC-PID rule has been proposed by Lee and Choi [49]. Using fixed integral and derivative times for controller tuning parameters may not be good for some processes.

In this chapter a simple multiloop controller tuning method is proposed, which utilizes a trial and error tuning method for SISO processes (Seborg et al. [77]). The trial-and -error method determines the controller gain, the integral time and the derivative time sequentially. Since it uses the continuous cycling data at each step, we call it the iterative continuous cycling(ICC) method. As shown later, the ICC method is generally superior to the Ziegler-Nichols tuning for first order

plus time-delay processes. Since the method determines one PID setting at each step, it can be applied to design multiloop systems using the Nyquist array method.

The modified sequential closing method (Chiu and Arkun [16]) and the independent design method (Skogestad and Morari [87]) can handle performance robustness. When uncertainty models are available, such controller design methods can be very effective. However, uncertainty models are very hard to obtain. Simple frequency-independent uncertainty bounds will result in controllers that are too conservative (McDonald et al.[60]). Hence, we believe it is more practical to design a control system for multivariable processes that have specified gain and phase margins instead of finding uncertainty models and designing a control system with a specified gain margin (e.g., 2.0).

7.2 Multiloop and Multivariable Controller Design for RTP Temperature Control

Multiloop PI control and quadratic dynamic matrix control(QDMC) algorithms are applied here for the temperature uniformity control of rapid thermal processing(RTP). Input and output variable transformations are done based on the singular value decomposition(SVD) of the process gain matrix and the μ-interaction measure analysis of compensated transfer function matrix. Multiloop PI control with transformation of variables shows improvement in transient responses to set point changes compared with the multiloop PI control without variable transformations and QDMC. Further studies on systematic tuning method for this multiloop PI control is needed to reduce the offsets and to obtain the uniform transient responses.

One major concern in single wafer RTP is the temperature non-uniformity across the wafer surface, especially for larger wafers. This has prompted a closer look at the design of RTP reactors and the synthesis of controllers for maintaining temperature uniformity. This section investigates the control of temperature uniformity across a wafer in a single wafer reactor. Temperatures at three different radii on the wafer are measured and controlled using the power supplied to three rings of lamps as the manipulated variables. Thermocouple-instrumented wafers have been tested in order to provide accurate temperature measurements for comparison with pyrometers that are normally used in rapid thermal chemical vapor deposition (RTCVD).

This is a multivariable input-output problem. Several papers have been published on multivariable control as opposed to multiloop single input single output (SISO) control. Schaper et al.[73] derived a first order model from first principles and then fit parameters to the model using empirical data. They then determined controller parameters using IMC tuning methods and employed gain scheduling to account for the nonlinearity in the process. Breedijk et al.[9] have demonstrated a novel idea of using nonlinear QDMC for real-time control. Their model derived from the first principles improved the conditioning of the control system. Cho et al.[17] have also performed temperature uniformity control studies using multiloop SISO and QDMC controllers. In this work, several multivariable control techniques are evaluated and compared with multiloop SISO control for a rapid thermal processor.

The rapid thermal processor consists of a three lamp zone illuminator assembly over a single wafer. The three lamp zones consist of an outer 24 lamp ring,

a middle 12 lamp ring and a central lamp. Temperatures at three radially independent distances from the center of the wafer are measured using a thermocouple-instrumented wafer. The power inputs to the three lamp zones are independent of each other and are the manipulated variables that are used to maintain temperature uniformity on the wafer. Figure 2.7 gives a schematic of the wafer-lamp assembly.

The open-loop responses of the thermocouple wafer temperatures can be obtained by step changes in each of the three lamp powers. Data analysis indicates that the process can be described by a first-order plus time delay transfer function matrix. However, a steady increase of the wafer temperatures mainly caused by the quartz window heating makes it difficult to obtain the steady state temperature, which is required for model identification. Values of the process gain, time constant, and time delay vary widely as the operating temperature changes. The influence of the chamber pressure, plasma power, and gas flow rate variables on the wafer temperature also causes the value of the each lamp power supply setting to deviate from prespecified values. The nonlinearity of the system was demonstrated previously for a RAPRO RTP reactor[12], and Cho and Kailath[18] showed that the values of the gain and the time constant for a RTP reactor, similar to the one used in this study, decreased monotonically as operating temperature increased (they were also affected by chamber pressure).

Closed-loop identification of the process model under proportional control was used(see Chapter 4). Diagonal elements of the transfer function matrix were calculated by matching the closed-loop gain and time constant data to an analytical expression. The process gains of the off-diagonal elements were calculated using the

same method as for the diagonal elements. The time constants were obtained by fitting closed-loop data with the closed-loop transfer functions.

Because this reactor could be used for remote plasma-enhanced CVD, a set point change of 375°C was made from room temperature to obtain an input-output transfer function matrix; the input variables u_1, u_2, and u_3 are the power levels for the edge, middle and center zone lamps, respectively, and the output variables T_{c1}, T_{c2}, and T_{c3} are thermocouple temperatures corresponding to the locations shown in Figure 2.7.

$$\begin{pmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \end{pmatrix} = G(s) \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{7.1}$$

where

$$G(s) = \begin{pmatrix} \frac{3.38\exp(-1.5s)}{38.6s+1} & \frac{2.50\exp(-2.5s)}{42.6s+1} & \frac{0.953\exp(-3.2s)}{153.3s+1} \\ \frac{3.20\exp(-1.7s)}{39.1s+1} & \frac{2.38\exp(-2.6s)}{44.0s+1} & \frac{0.986\exp(-3.0s)}{105.2s+1} \\ \frac{3.13\exp(-1.5s)}{47.3s+1} & \frac{2.33\exp(-2.3s)}{43.8s+1} & \frac{1.054\exp(-2.4s)}{62.9s+1} \end{pmatrix}$$

7.2.1 Steady state analysis

The steady state transfer function matrix was analyzed using the relative gain array (RGA) and the condition number(κ) of the system. The RGA and the κ for a system are determined as follows:

For the steady state gain matrix, G(0), the RGA is given by

$$RGA = G(0) \otimes [G(0)^{-1}]^T$$

where the '$\otimes$' operator signifies element by element matrix multiplication operation. The singular value decomposition of G(0) is given by

$$G(0) = U\Sigma V^T$$

where matrix U contains the left singular vectors of G(0), matrix Σ contains the singular values of G(0) arranged along the diagonal, and matrix V contains the right singular vectors of G(0). The κ of the system is the ratio of the largest singular value to the smallest singular value. The higher the κ, the more ill-conditioned the system.

Results for RGA, SVD and condition number for the 3x3 steady state matrix are shown below. The absence of elements close to 1.0 in the RGA indicates the highly coupled nature of the system. The large κ resulting from a significant difference between the largest and smallest singular values is also indicative of the ill-conditioning of the system.

$$RGA = \begin{pmatrix} 210.3 & -211.2 & -1.853 \\ -390.9 & 406.6 & -14.65 \\ 181.6 & -194.4 & 13.79 \end{pmatrix}$$

SVD gives

$U\Sigma V^T =$

$$\begin{pmatrix} -0.5989 & 0.6887 & 0.4086 \\ -0.5709 & -0.0094 & -0.8210 \\ -0.5616 & -0.7250 & 0.3988 \end{pmatrix} \cdot \begin{pmatrix} 7.1959 & 0.0000 & 0.0000 \\ 0.0000 & 0.1209 & 0.0000 \\ 0.0000 & 0.0000 & 0.0039 \end{pmatrix}$$

$$\cdot \begin{pmatrix} -0.7795 & -0.5787 & -0.2398 \\ 0.2356 & 0.0838 & -0.9682 \\ 0.5804 & -0.8112 & 0.0710 \end{pmatrix}$$

$$\kappa = 7.1959/\ 0.0039 = 1845$$

7.2.2 Interaction analysis

A multiloop PI controller is the first choice to control a multi-input multi-output process. For this controller to be successful the process should be diagonally dominant for wide range of frequencies. A dynamic interaction measure(known as the μ-interaction measure), which indicates diagonal dominance of the process investigated, has been proposed by Grosdidier and Morari[34] as :

$$\mu^{-1}\big(E(j\omega)\big)$$

where

$$\begin{aligned} E(j\omega) &= \big(G(j\omega) - \tilde{G}(j\omega)\big)\tilde{G}^{-1}(j\omega) \\ \tilde{G}(j\omega) &= diag\big(G(j\omega)\big) \end{aligned}$$

If the μ-interaction measure is greater than unity, it is diagonally dominant at that specific frequency and the bandwidth of the decentralized multiloop control system can be enlarged to the frequency to which the process maintains diagonal dominance. Since it is a sufficient condition, we should be cautious in using the μ-interaction measure, i.e. there may exist a good multiloop controller which violates the μ-interaction measure criterion.

The solid line of Figure 7.1 shows the μ-interaction measure for $G(s)$. Since the μ-interaction measure is less than unity through the frequencies shown, one can

guess that fair amount of interaction exists in the process and the multiloop control would be problematic. Large values of the elements of the RGA also indicate the difficulty of the multiloop process control.

A simple compensator can be implemented as a part of the controller to reduce the interaction. Decouplers based on the steady-state gain matrix can be used for this purpose. The dotted line of Figure 7.1 shows the μ-interaction measure for the process with steady-state decoupler, $G(s)G(0)^{-1}$. As the frequency increases, the value of μ-interaction measure falls below unity at around 0.001 rad/sec. Hence a fast closed-loop response would not be obtained based on this steady-state decoupler. Furthermore, since the condition number of the process gain matrix is very high, the controllers using $G(0)^{-1}$ in their design are highly sensitive to model mismatch and numerical computation.

To reduce problems of the above static decoupler, decoupling based on the SVD of the process gain matrix is investigated. As in Lau et al.[46], one can obtain an approximate structural compensator which decomposes the process as

$$y_p = K_1 G(s) K_2 u_p$$

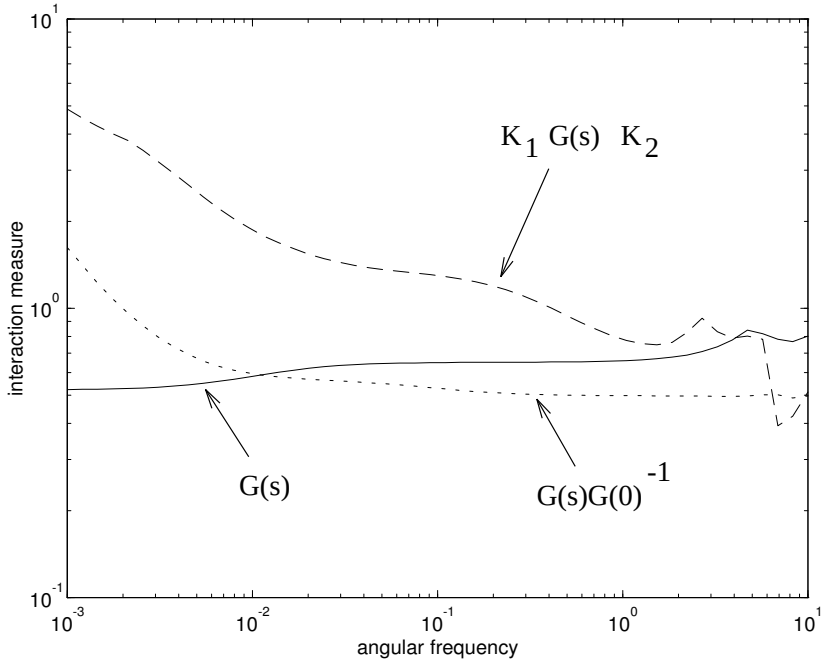


Figure 7.1: μ-interaction measure for several multiloop control systems

where

$$
\begin{aligned}
y_p &= K_1 y \\
u_p &= K_2^{-1} u \\
y &= \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \end{bmatrix} \\
u &= \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \\
K_1 &= \begin{pmatrix} 0.333 & 0.333 & 0.333 \\ -0.500 & 0.000 & 0.500 \\ 0.250 & -0.500 & 0.250 \end{pmatrix}
\end{aligned}
$$

and

$$
K_2 = \begin{pmatrix} 0.780 & -0.236 & 0.580 \\ 0.579 & -0.084 & -0.811 \\ 0.240 & 0.968 & 0.071 \end{pmatrix}
$$

The matrix K_1 is an approximation to matrix U^T (or U^{-1} and K_2 is equal to V $\left(\text{or } (V^T)^{-1}\right)$. Hence, at steady-state, y_p and u_p variables are approximately decoupled. Although the condition number of the process gain matrix is very high, there are no large elements in matrices K_1 and K_2. The matrix K_1 shows that one of the new outputs in y_p is the the average of the three temperatures and the other two are differences between the temperatures. These new transformed outputs give improved transient tracking and uniformity. The RGA for the compensated transfer function, $K_1 G(0) K_2$ is now

$$\begin{pmatrix} 0.998 & 0.001 & 0.001 \\ 0.001 & 0.999 & 0.000 \\ 0.001 & 0.000 & 0.999 \end{pmatrix}$$

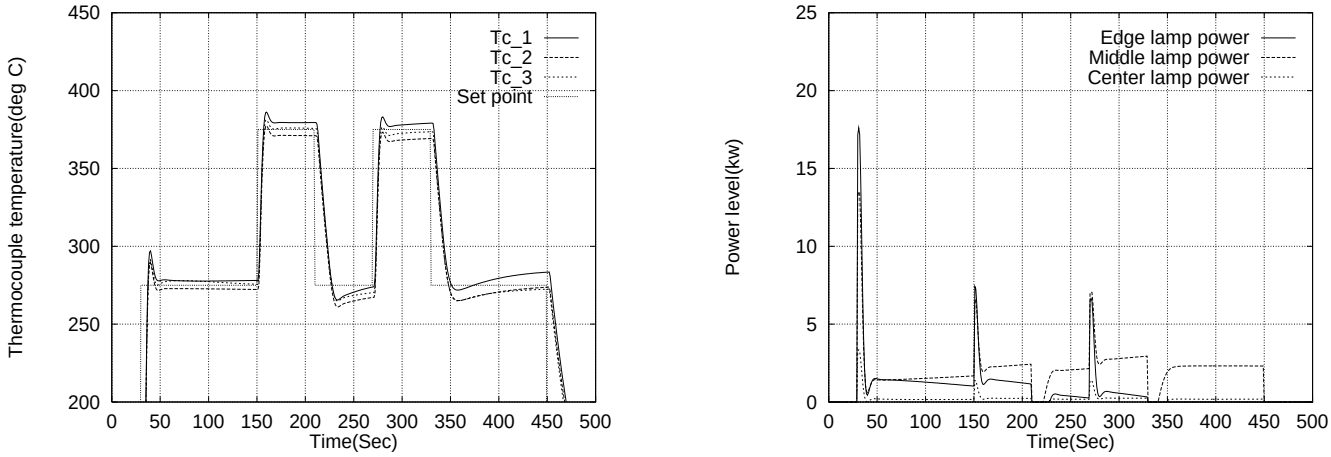
The μ-interaction measure for this compensated transfer function matrix is also shown in Figure 7.1(dashed line). The μ-interaction measure for this design is far better than the previous designs. However, one can see that it falls down rapidly below unity near the angular frequency $\omega = 0.41$ rad/sec. This would restrict the control performance. Implementation of the dynamic compensator is not considered further because the simplicity of the multiloop PI control strategy would not be maintained. If more advanced features such as dynamic compensation is required, one can implement more elaborate controllers such as QDMC.

7.3 Control Results

7.3.1 Case 1: Multiloop PI control

PI controller settings were obtained through internal model control(IMC) based PI tuning[77]. Simulation results are shown in Figure 7.2.

Figures 7.3 and 7.4 show experimental results for (a) the thermocouple wafer temperature responses for prespecified set point changes, and (b) the controller output levels, with sampling rates 1 and 10 Hz, respectively. Since the model was identified at around 400°C , the temperature responses for the set point change to 275°C were not as good as those for the set point change to 375°C , and the performance of the controller with faster sampling rate(10 Hz) is better than that with the slower sampling rate(1 Hz), with less overshoot. beginfigure[h]



(a) (b)

Figure 7.2: Multiloop PI control simulation results

7.3.2 Case 2: Multiloop PI control with variable transformation

Multiloop PI control system for the process compensated structurally with the SVD of the process gain matrix has been simulated. The PI controller settings were obtained by trial and error. One set of closed-loop responses is shown in Figure 7.5. Similar settling times to Case 1 but with smoother transient responses are obtained. A slow drift near the set point are also found as in the multiloop PI controllers. They may be due to large interaction and poor controller settings. A systematic fine tuning method is needed.

7.3.3 Case 3: QDMC

Simulation has been done for the model using a QDMC algorithm[28] based on quadratic programming combined with model predictive control to control wafer temperature uniformity. Figure 7.6 shows slightly better responses compared to

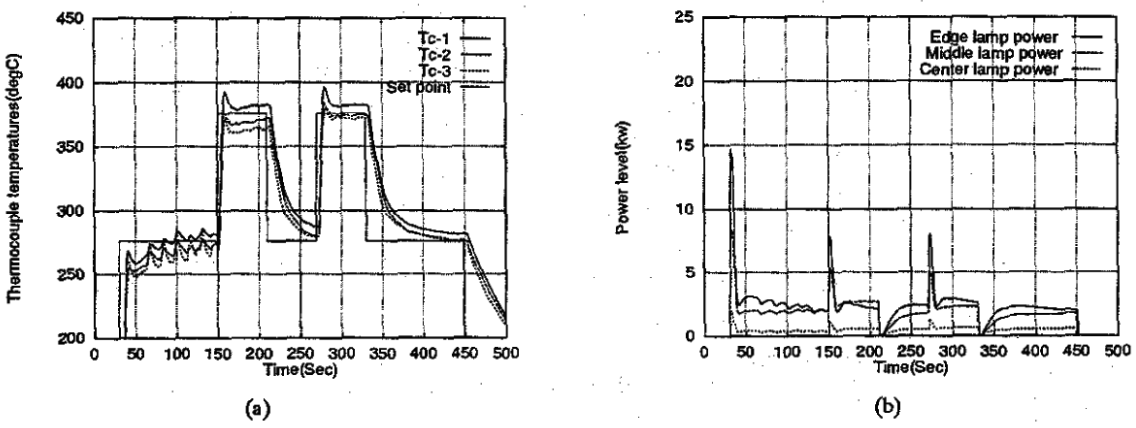


Figure 7.3: Multiloop PI control experimental results; sampling rate=1 Hz

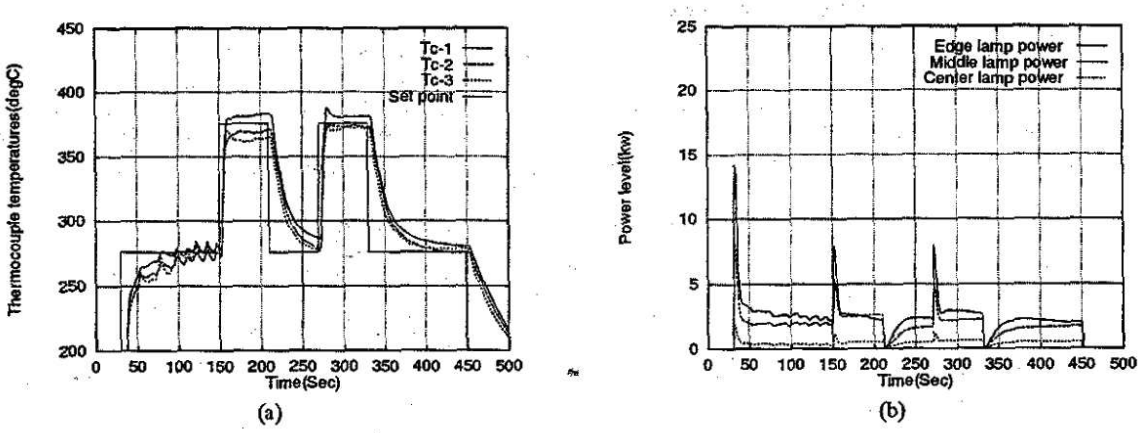


Figure 7.4: Multiloop PI control experimental results; sampling rate=10 Hz

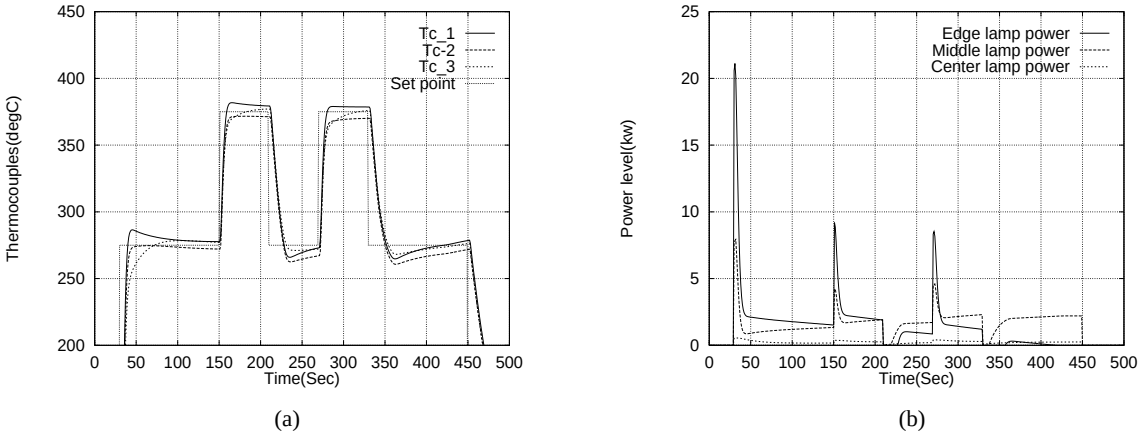


Figure 7.5: Multiloop PI control with variable transformation simulation results

cases 1 and 2. The controller outputs show larger movements for QDMC than for PI controllers, so when there is a large set point change, the controller performance may not be good if the controller output is bounded(constrained). The upper power limits in Figure 7.6 are 24 kw, 12 kw, and 2 kw for edge, middle, and center lamp zone, respectively.

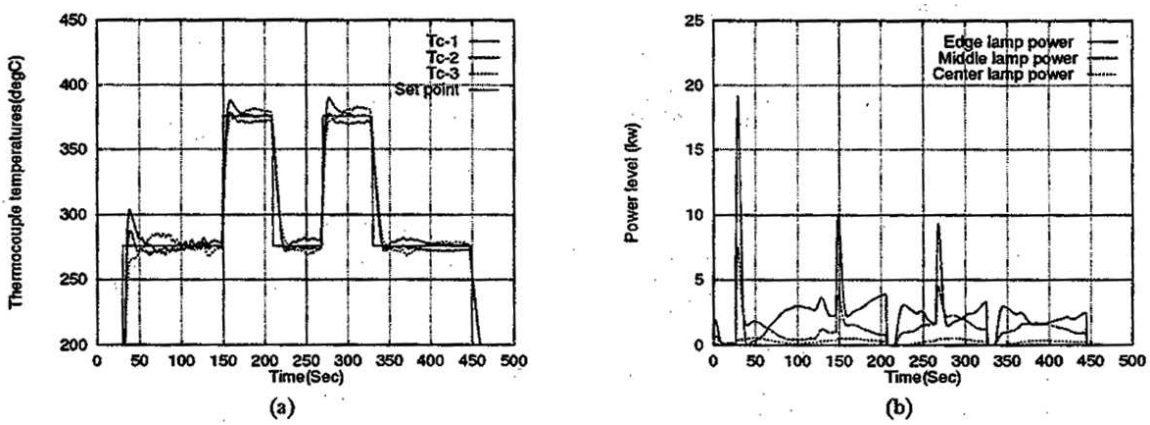


Figure 7.6: Simulated QDMC results; sampling rate=1 Hz

7.4 Conclusions (Multiloop vs. Multivariable Control)

For temperature set point changes used in RTP, multiloop PI controllers performed nearly as well as the QDMC controller. However, there are temperature differences of 10–20°C at the 375°C set point change. This is due to the strong interaction between the edge and middle ring zone. More extensive system identification will be done in the future, at various temperature ranges, to obtain optimum PI controller tuning parameters for a given operating temperature. Closed-loop identification is required for various operating regions to get the best controller parameters for each region. To overcome the system nonlinearity, a gain scheduling multiloop PI control algorithm can be employed. Improved on-line temperature measurement techniques to complement the pyrometers also need to be developed.

7.5 SISO iterative continuous cycling (ICC) tuning

A typical procedure to tune PID controllers in field is as follows (trial and error tuning in Seborg et al.[77]):

Step 1: Eliminate the integral and derivative actions by setting the integral time at its maximum and the derivative time at its minimum. increase the controller gain until continuous cycling occurs. Set the controller gain at half of the value.

Step 2: Decrease the integral time until continuous cycling occurs. Set the integral time at 3 times the value.

Step 3: Increase the derivative time until continuous cycling occurs. Set the derivative time at one third of the value.

This approach that we call iterative continuous cycling(ICC) method is extended to design multiloop PI controllers. Step 2 is slightly modified as the integral time is set at 2.5 times its ultimate value. Comparison of closed-loop regulator performances are shown in Figure 7.7 for PI controllers and various time delays.

We can see that the control performances of ICC tuning for load changes are better than those of Ziegler-Nichols and ITAE tuning. gain and phase margins as a function of time delays are shown in Figure 7.8. The ICC method has gain margin of about 1.7 throughout the time delay range (0.1 and 1.0) and phase margins above 30. We can use the ICC tuning method as a model-based SISO PI controller tuning method because the calculation of the continuous cycling data for a process model is not difficult.

7.6 Stability conditions for decentralized control systems

The SISO ICC tuning method can be extended to multiloop controller tuning. It is necessary to find the largest controller gains for each loop, which assures stability. There are several methods to find real parameters for which the system remains stable. One elegant method is the Nyquist array method. Consider a process with n inputs and n outputs:

$$y(s) = G(s)u(s) \tag{7.2}$$

where G(s) is an nxn process transfer function matrix, y is an n output vector and u is an n input vector. The stability of the closed-loop system for a diagonal controller

$$C(s) = diag\{c_1(s), c_2(s), \ldots, c_n(s)\}$$

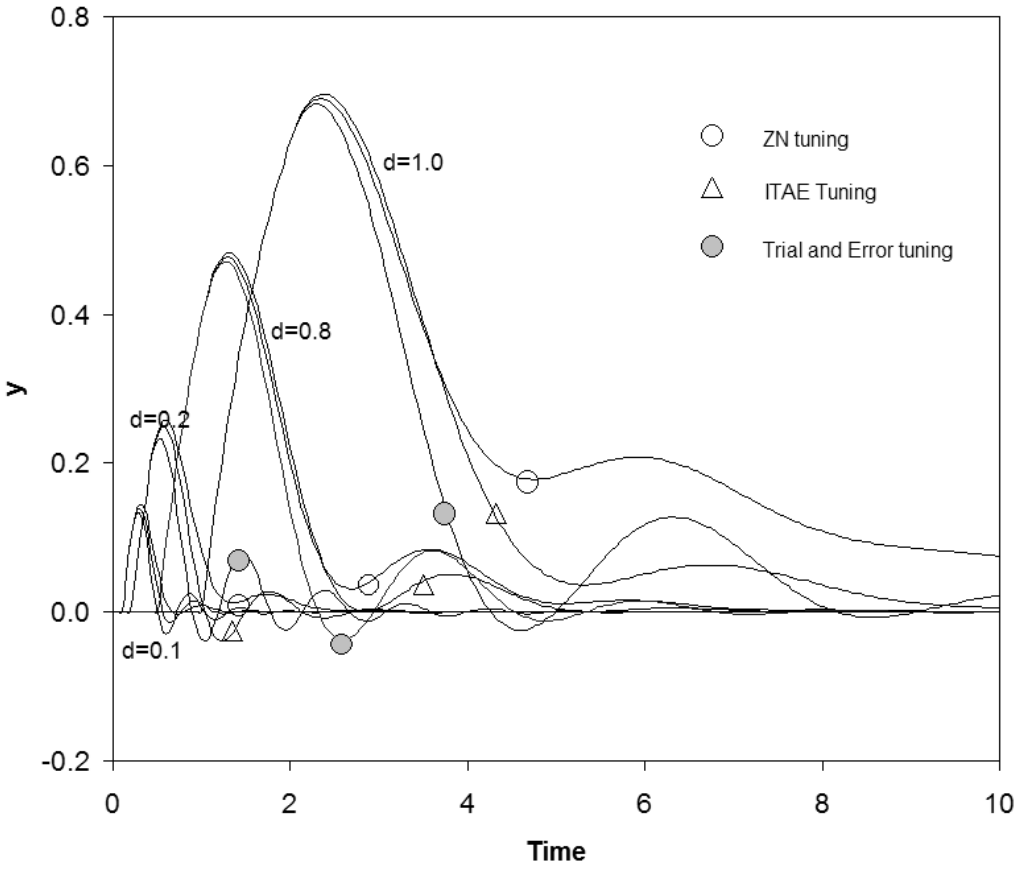
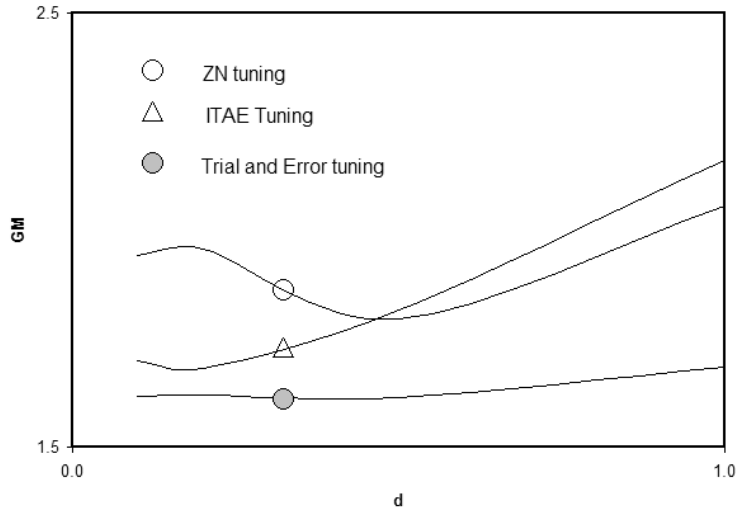
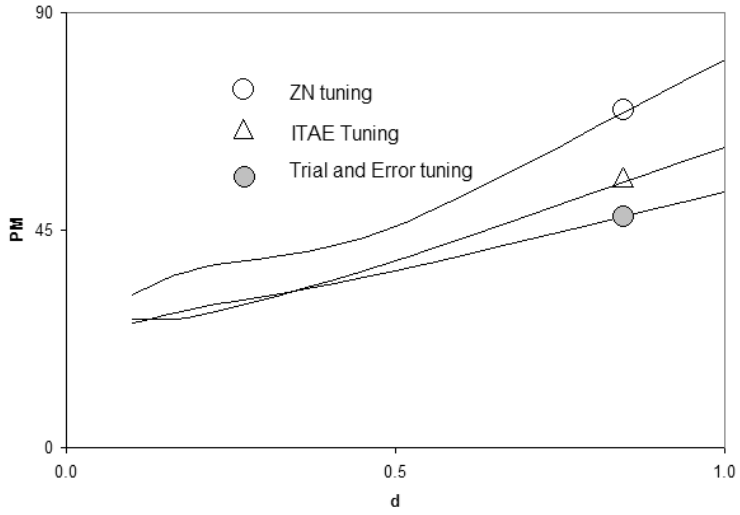


Figure 7.7: Load responses of three tuning methods (PI controllers) for the process dynamics, $\exp(-ds)/(s+1)$.



(a)



(b)

Figure 7.8: (a) Gain and (b) phase margins for three tuning methods (PI controllers) for the process, $exp(-ds)/(s+1)$.

which is designed for the diagonal subsystem of $G(s)$

$$\widetilde{G}(s) = diag\{g_{11}(s), g_{22}(s), \ldots, g_{nn}(s)\}$$

is to be determined. In the Nyquist array method (Rosenbrock [36]), Gershgorin circles superimposed on the diagonal elements of Nyquist array can be used to determine if the closed-loop system remains stable in spite of interaction terms.

Grosdidier and Morari[34] considered the diagonal closed-loop transfer function

$$\begin{aligned} \widetilde{H}(s) &= \widetilde{G}(s)C(s)[I + \widetilde{G}(s)C(s)]^{-1} \\ &= diag\{\widetilde{h}_i(s)\} \end{aligned} \tag{7.3}$$

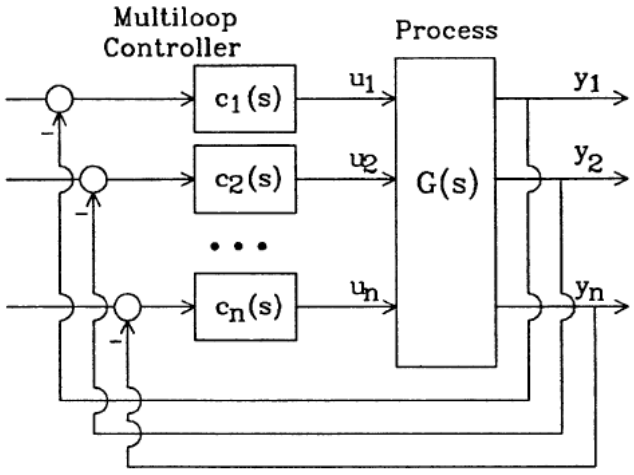
which guarantees the stability of the closed-loop system

$$H(s; \widetilde{H}(s)) = G(s)C(s)[I + G(s)C(s)]^{-1}$$

Here $H(s; \widetilde{H}(s))$ is the closed-loop transfer function matrix where the diagonal controller $c(S)$ is calculated from $\widetilde{H}(s)$. From equation 7.3, $C(s)$ can be calculated except when $\widetilde{G}(s)$ is identically singular. Grosdidier and Morari[34] presented constraints on the choice of $\widetilde{H}(s)$ that guarantee the stability of the closed-loop system $H(s; \widetilde{H}(s))$.

New stability conditions on $\widetilde{H}(s)$ are presented below. As in Grosdidier and Morari [34], we use the matrix

$$E(s) = [G(s) - \widetilde{G}(s)]\widetilde{G}^{-1}(s)$$



(a)

Figure 7.9: Multiloop control system

to represent relative error when the full system $G(s)$ is approximated by the diagonal subsystem $\widetilde{G}(s)$. Let $N\{k, g(s)\}$ be the net number of clockwise encirclements of the point $(k, 0)$ by the image of the Nyquist contour under a scalar transfer function $g(s)$ (it is assumed that the image forms closed curves; Desoer and Wang[21]). Using the multivariable Nyquist criterion (Postlethwaite and MacFarlane[66]) obtained following stability condition for the control system in Figure 7.9.

Theorem 1. (Grosdidier and Morari [34]). *Assume that G(s) and* $\widetilde{G}(s)$ *have the same number of right half-plane poles and that* $\widetilde{H}(s)$ *is stable. Then the closed-loop system* $H(s; \widetilde{H}(s))$ *is stable if*

$$N\{0, det(I + E(s)\widetilde{H}(s))\} = 0$$

In the above theorem, they unified many interaction measures and derived

the μ-interaction measure. For example, under the assumptions of Theorem 1, the closed-loop system is stable if

$$|\widetilde{h}_i(j\omega)| < \mu^{-1}(E(jw)) \tag{7.4}$$

for any i and real ω. Here $\mu(\cdot)$ is the structured singular value with respect to the diagonal structure of $\widetilde{H}(s)$ ([23]. From the $\mu^{-1}(E(j\omega))$ versus frequency (ω) plot, the decentralized transfer function $\widetilde{H}(s)$ can be easily designed. The $\mu^{-1}(E(j\omega))$ is used as the dynamic interaction measure (μ-interaction measure). the structured singular value of equation 7.4 is the tightest bound on the magnitude of $\widetilde{H}(j\omega)$, but its computation is rather complex (Fan and Tits [26]). So the following upper bound of $\mu(E(j\omega))$ is often used (Maciejowski [58]):

$$\begin{aligned} \mu(E(j\omega)) \qquad & inf_s \sigma_{\max}(SE(j\omega)S^{-1}) \\ & <= \rho(abs(E(j\omega))) \end{aligned} \tag{7.5}$$

where S is any real positive diagonal matrix, $\sigma_{\max}(\cdot)$ and $\rho(\cdot)$ are the maximum singular value and the spectral radius of a matrix, respectively, and $abs(\cdot)$ is a matrix with all its elements replaced by their magnitudes. When $n <= 3, \mu(E(j\omega)) = inf_s \sigma_{max}(SE(j\omega)S^{-1})$. The bound $\rho^{-1}(abs(E(j\omega)))$ is invariant for input and output scaling and is closely related to conditions for generalized diagonal dominance in the Nyquist array method (Grosdisdier and Morari [34]).

Condition 7.4 is very useful, but it is rather conservative because the phase information in $\widetilde{H}(j\omega)$ is not utilized. In the case of identical decentralized closed-loop transfer functions ($\widetilde{H}(s) = \widetilde{h}(s) \cdot I$), the right-hand-side of equation 7.4 can be

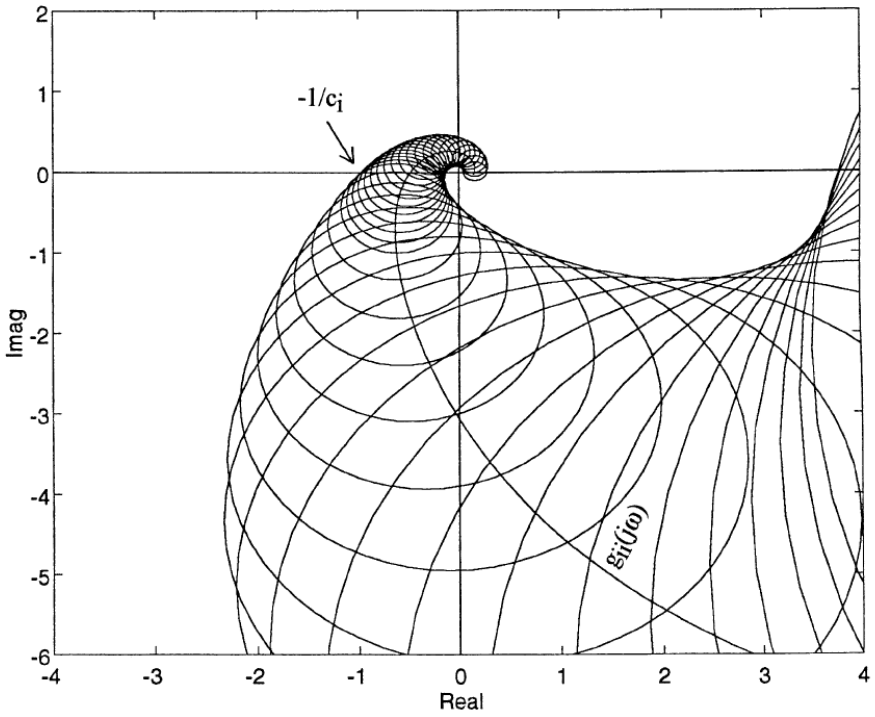


Figure 7.10: Stability circles from the μ-interaction measure (equation 7.14)

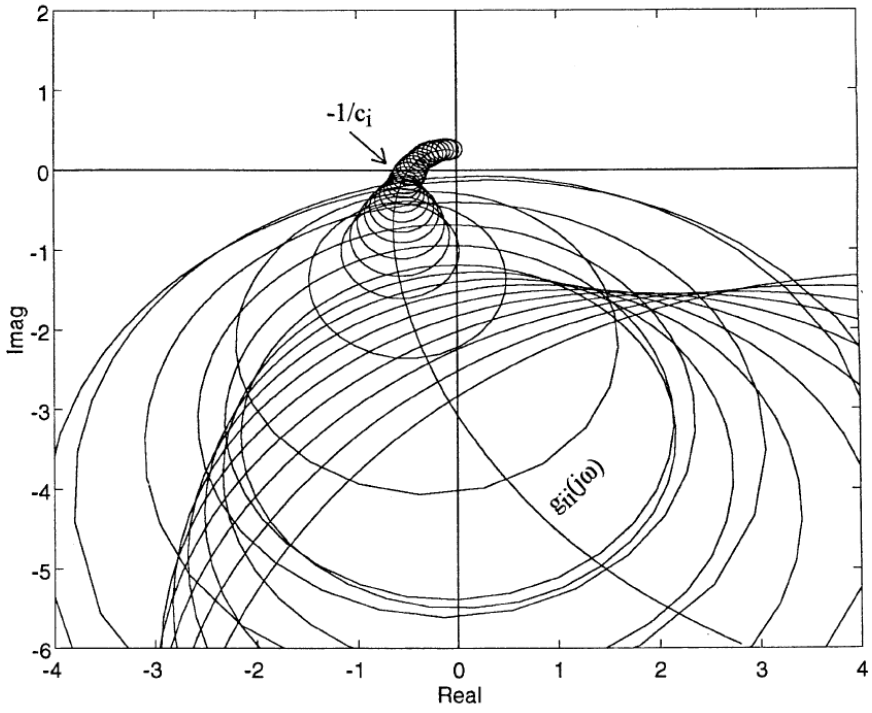


Figure 7.11: Stability curves from equation 7.16

replaced by $\rho^{-1}(E(j\omega))$, which is usually much greater than $\mu^{-1}(E(j\omega))$. Here, for less conservative stability conditions, we split $\widetilde{H}(s)$ into two parts as

$$\widetilde{H}(s) = \widetilde{H}_{\mathrm{B}}(s)[I + \widetilde{H}_{\mathrm{M}}]$$

where $\widetilde{H}_{\mathrm{B}}(s)$ is for a known fixed part and $\widetilde{H}_{\mathrm{M}}(s)$ is for a design part. The following stability conditions are obtained (Lee et al.[50]).

Lemma 1. *Assume that $G(s)$ and $\widetilde{G}(s)$ have the same number of right half-plane poles and that $H(s)$ and $\widetilde{H}(s)$ is stable. Then the closed-loop system $H(s;\widetilde{H}(s))$ is stable if*

$$N\{0, det(I + E(s)\widetilde{H}_{\mathrm{B}}(s))\} = 0$$

and

$$N\{0, det(I + [I + E(s)\widetilde{H}_{\mathrm{B}}(s)]^{-1}E(s)\widetilde{H}_{\mathrm{B}}(s)\widetilde{H}_{\mathrm{M}}(s))\} = 0$$

The assumption of the lemma is such that the number and location of unstable poles of $G(s)$ and $\widetilde{G}(s)$ are the same. it is satisfied trivially for open-loop stable systems or for systems with unstable poles that appear with a special structures in the model transfer function matrix. Since most practical processes are open-loop stable, the assumptions of the above lemma are not too restrictive. We may also derive stability condition based on the number of zeros as in Skogestad and Morari [87] and Postlethwaite and Foo [42], which are imposed on the decentralized sensitivity functions instead of the decentralized closed-loop transfer functions.

Stability conditions on $\tilde{H}_B(s)$ and $\tilde{H}_M(s)$ can be derived from Lemma 1 in the same way that Grosdidier and Morari [34] obtained the μ-interaction measure

from Theorem 1. Appropriate $\tilde{H}_B(s)$ will provide new stability conditions. A transfer function $\frac{1}{2}\tilde{H}(s)$ is chosen for $\tilde{H}_B(s)$ to obtain conditions for loop failure tolerance as well as stability, because it is the average of a closed-loop transfer function $\tilde{H}(s)$ and 0. Assuming $\tilde{H}_B(s) = \frac{1}{2}\tilde{H}(s)$ leads to the following stability conditions[50].

Theorem 2 *Assume that G(s) and $\tilde{G}(s)$ have the same number of right half-plane poles and that $\tilde{H}(s)$ and $\hat{H}(s)$ are stable. Then the closed-loop system $H(s;\hat{H}(s))$ is stable if*

$$N\{0, det(I + \frac{1}{2}E(s)\tilde{H}(s))\} = 0 \tag{7.6}$$

and

$$|\frac{\hat{h}_i(j\omega)}{\tilde{h}_i(j\omega)} - \frac{1}{2}| < \mu^{-1}([I + \frac{1}{2}E(j\omega)\tilde{H}(j\omega)]^{-1}xE(j\omega)\tilde{H}(j\omega)) \tag{7.7}$$

for any i and real ω.

Condition 7.7 can be applied to the Nyquist array method, generalizing the Gershgorin circles. To obtain better stability curves of equation 7.7, $\tilde{H}(s)$ may be adjusted iteratively or through some other methods. Application of equation 7.7 to the Nyquist array method is shown in the following section.

Theorem 3. *Assume that G(s) and $\tilde{G}(s)$ have the same number of right half plane poles and that $\tilde{H}$(s) is stable. Then the closed-loop system H(s;$\tilde{H}$(s)) is stable if*

$$N\{0, det(I + \frac{1}{2}E(s)\tilde{H}(s))\} = 0$$

and

$$\mu([I + \frac{1}{2}E(j\omega)\tilde{H}(j\omega)]^{-1}E(j\omega)\tilde{H}(j\omega)) < 2 \tag{7.8}$$

for any real ω.

Equation 7.8 is complex but $\tilde{H}(s)$, which satisfies Theorem 3, has very useful properties as follows.

Theorem 4. *Assume that G(s) is stable and that $\tilde{H}(s)$ satisfies Theorem 3. Then the closed-loop system H(s;$\hat{H}$(s)) is stable and loop failure tolerant if, for any i,*

$$\hat{h}_i(s) = \frac{\tilde{\tau}_i s + 1}{\hat{\tau}_i s + 1} \tilde{h}_i(s) \tag{7.9}$$

where $\hat{\tau}_i \geq \tilde{\tau}_i \geq 0$. Further it is loop failure tolerant if

$$Re(\tilde{h}_I(j\omega)) \leq 1 \tag{7.10}$$

for any real ω.

The μ-interaction measure of equation 7.4 also requires condition 7.10 for loop failure tolerance. The above theorem provides an easy way to adjust the bandwidth of $\tilde{H}(s)$ of Theorem 3. Specifically, it can be used to generate slower stable decentralized controllers that are loop failure tolerant. For example, if $\tilde{H}(s) = diag\{1/(\hat{\tau}_i s + 1)\}$ for $\hat{\tau}_i \geq \tilde{\tau} > 0$ results in a stable closed-loop system with loop failure tolerance.

Equation 7.8 is less conservative than the μ-interaction measure of equation 7.4 if the structured singular value is approximated with either upper bounds of equation 7.5 or any induced matrix norm. It can be proved as, for the one upper bound of equation 7.5,

$$\begin{aligned}\mu([I &+ \frac{1}{2}E(j\omega)\tilde{H}(j\omega)]^{-1}E(j\omega)\tilde{H}(j\omega)) \\ &\leq \sigma_{\max}(S[I+\frac{1}{2}E(j\omega)\tilde{H}(j\omega)]^{-1}E(j\omega)\tilde{H}(j\omega)S^{-1} \\ &\leq \frac{\sigma_{\max}(SE(j\omega)S^{-1}\tilde{H}(j\omega))}{1-\frac{1}{2}\sigma_{\max}(SE(j\omega)S^{-1}\tilde{H}(j\omega))} \\ &< 2\end{aligned}$$

if $\sigma_{\max}(\tilde{H}(j\omega)) < \sigma^{-1}_{rmmax}(SE(j\omega)S^{-1})$.

For a stable 2 x 2 system the stability condition of Theorem 1 becomes (Grosdidier and Morari [34]

$$N\{1, z(s)\} = 0 \tag{7.11}$$

where $z(s) = \tilde{h}_1(s)\tilde{h}_2(2)g_{12}(s)g_{21}(s)/(g_{11}(s)g_{22}(s))$. The μ-interaction measure means $|z(j\omega)| < 1$ for any real ω. The regions of condition 7.8 and its approximation are shown in Figure 7.12. Equation 7.11 is necessary and sufficient, but loop failure tolerance is not guaranteed.

Corollary 1. *Assume that G(s) is stable. hen there exists a stable decentralized controller C(s) with integral action and loop failure tolerance if*

$$\rho(E(0)) < 2 \tag{7.12}$$

it and

$$\mu([I+\frac{1}{2}E(0)]^{-1}E(0)) < 2 \tag{7.13}$$

The above corollary is a sufficient condition for integral controllability(Grosdidier et al.[34]) and for loop failure tolerance of the closed-loop system and is expressed in

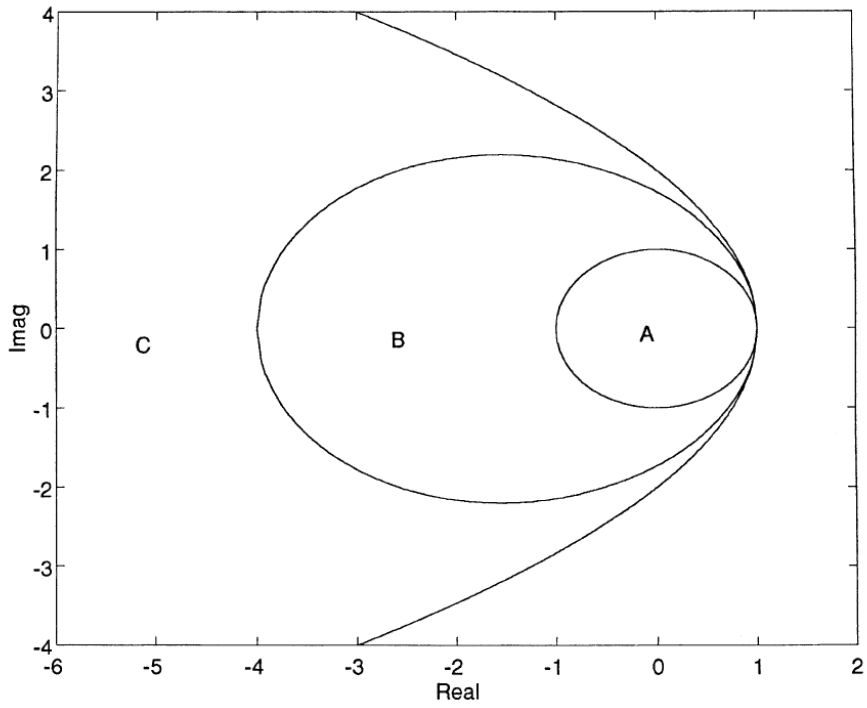


Figure 7.12: Stability regions for $z(s) = \tilde{h}_1(s)\tilde{h}_2(2)g_{12}(s)g_{21}(s)/(g_{11}(s)g_{22}(s))$. (A) the μ-interaction measure, (B) condition 7.8 with an upper bound of μ value, (C) condition 7.8

terms of the steady-state gains of $G(s)$. It should be remarked that $\tilde{G}(s)$ need not be diag$g_{ii}(s)$. The first- or second-order approximation of $g_{ii}(s)$ can be used. This choice with the IMC-PID tuning rule results in PI or PID-type multiloop controllers directly. The reciprocal of diagonal elements of $G^{-1}(s)$ is another alternative for $\tilde{G}(s)$, which may give less conservative stability conditions.

7.7 Graphical interpretation of the stability theorems

Theorems 1 and 2 can be used to find real stability gains graphically. it is shown in Grosdidier and Morari [34] that condition 7.4 improves the Gershgorin bands of the conventional Nyquist array method. Condition 7.7 can be used to further improve the gains of condition 7.4.

When $(-1/c_i, 0)$ is outside the curves;

$$g_{ii}(j\omega)(1+\mu^{-1}(E(j\omega)e^{j\theta})) \qquad \theta \in [0, 2\pi] \tag{7.14}$$

then condition 7.4 is satisfied. These circles are shown in Figure 7.10.

For a given $\tilde{H}(s)$ satisfying equation 7.6, condition 7.7 is satisfied when $\hat{h}_i(j\omega)$ is within circle

$$\tilde{h}_i(j\omega)(\frac{1}{2}+\beta(\omega)e^{j\theta}) \qquad \theta \in [0, 2\pi] \tag{7.15}$$

where, $\beta(\omega) = \mu^{-1}([I+\frac{1}{2}E(j\omega)\tilde{H}(j\omega)]^{-1}E(j\omega)\tilde{H}(j\omega))$, or equivalently when $(-1/c_i, 0)$ is outside the curve

$$g_{ii}(j\omega)\{1-\frac{1}{\tilde{h}_i(j\omega)(\frac{1}{2}+\beta(\omega)e^{j\theta})}\} \tag{7.16}$$

for $\beta(\omega) < 0.5$. Equation 7.16 is derived by applying conformal mapping techniques of complex functions to $\hat{h}_i(s) = c_i g_{ii}(s)/(1 + c_i g_{ii}(s))$. These circles are shown in Figure 7.11.

The upper limit of the stable proportional gain c_i predicted by equation 7.16 is dependent on $\tilde{H}(s)$. Equation 7.16 improves the generalized Gershgorin band of equation 7.14 and it may be used iteratively. Theorems 3 and 4 can be used to check efficiently the loop failure tolerance of control systems designed by other tuning methods such as BLT and sequential closing. Corollary 1 can be used to determine if Theorems 2 and 3 can be applied.

7.8 Multiloop ICC tuning method

Multiloop PI controllers, $C(s) = K_1(I + K_2/s)$, are designed sequentially as in the SISO ICC method. The design procedure is as follows:

1. *Proportional gain design:* Design the proportional gain K_1 by setting $K_2 = 0$, using condition 7.4 graphically (equation 7.14). Then improve the gain once by applying condition 7.7(equation 7.16) using $\tilde{H}(s)$ with the previous proportional controllers. Set the proportional gains at half of the gains calculated also.

2. *Integral time design:* The characteristic equation for the multiloop PI control system is

$$det\{I + G(s)K_1(I + K_2/s\} = 0 \tag{7.17}$$

Since

$$\begin{aligned} \det \quad & \{I \quad + G(s)K_1(I + K_2/s)\} \\ & = \quad \det\{\mathrm{I} + \mathrm{G(s)K_1}\}\det\{\mathrm{I} + [\mathrm{I} + \mathrm{G(s)K_1}]^{-1}\mathrm{G(s)K_1K_2/s})\} \end{aligned} \tag{7.18}$$

we can design K_2 by applying the Nyquist array method to the rearranged process

$$[I + G(s)K_1]^{-1}G(s)K_1/s \tag{7.19}$$

whenever K_1 is the stable gain matrix. With the proportional gains of step 1 as K_1, design K_2 by applying condition 7.4 to the rearranged process 7.19. Set K_2 as 0.4 times the above value (integral times becomes 2.5 times larger).

3. *proportional gain adjustment* To ensure the gain margin of 2, we adjust the proportional gain again. We apply condition (6) to the process with integral action

$$G(s)(I + K_2/s) \tag{7.20}$$

with $\tilde{H}(s)$ designed in steps 1 and 2. Condition 7.7 is applied two times iteratively.

7.9 Simulation examples

Our method is compared with three other methods, the biggest log-modulus tuning(BLT) method of Luyben [57], the sequential autotuning method (Loh et al. [54]) and an independent design method of Lee and Choi [49], which is based on the SISO IMC-PID tuning. The last method utilizes IMC-PID tuning for each diagonal transfer function. Then it detunes the IMC-PID tuning by increasing the IMC filter

parameter λ. Since the IMC filter parameter affects only the proportional gain, the detuning is carried out by the Nyquist array method proposed in this work. Three 2 x 2 processes and one 4 x 4 process were selected from Luyben [57] as follows:

WB column:

$$G(s) = \begin{pmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix}$$

VL column:

$$G(s) = \begin{pmatrix} \frac{-2.2e^{-s}}{7s+1} & \frac{1.3e^{-0.3s}}{7s+1} \\ \frac{-2.8e^{(}-1.8s)}{9.5s+1} & \frac{4.3e^{-0.35s}}{9.2s+1} \end{pmatrix}$$

WW column:

$$G(s) = \begin{pmatrix} \frac{0.126e^{-6s}}{60s+1} & \frac{-0.101e^{-12s}}{(48s+1)(45s+1)} \\ \frac{0.094e^{(}-8s)}{38s+1} & \frac{-0.12e^{-8s}}{35s+1} \end{pmatrix}$$

A1 column:

$$G(s) = \begin{pmatrix} \frac{2.22e^{-2.5s}}{(36s+1)(25s+1)} & \frac{-2.94(7.9s+1)e^{-0.05s}}{(23.7s+1)^2} & \frac{0.017e^{-0.2s}}{(31.6s+1)(7s+1)} & \frac{-0.64e^{-20s}}{(29s+1)^2} \\ \frac{-2.33e^{(}-5s)}{(35s+1)^2} & \frac{3.46e^{-1.01s}}{32s+1} & \frac{-0.51e^{-7.5s}}{(32s+1)^2} & \frac{1.68e^{-2s}}{(28s+1)^2} \\ \frac{-1.06e^{(}-2.2s)}{(17s+1)^2} & \frac{3.511e^{-13s}}{(12s+1)^2} & \frac{4.41e^{-1.01s}}{16.2s+1} & \frac{-5.38e^{-0.5s}}{17s+1} \\ \frac{-5.73e^{(}-2.5s)}{(8s+1)(50s+1)} & \frac{4.32(25s+1)e^{-0.01s}}{(50s+1)(5s+1)} & \frac{-1.52e^{-2.8s}}{(43.6s+1)(9s+1)} & \frac{4.78e^{-1.15s}}{(48s+1)(5s+1)} \end{pmatrix}$$

The resulting multiloop controllers are shown in Table 7.1. For calculation of μ values the Osborne method in MATLAB Robust Control Toolbox was used. Closed-loop responses are shown in Figures 7.13 through 7.16 for step set-point changes. For the sequential autotuning method and the proposed method, set-point weighting on the P mode is set to zero (Loh et al. [54]). The BLT method shows a slow return to set point for 2 x 2 processes and it is oscillatory for the 4 x 4 process of A1 column. That is, detuning is high for 2 x 2 processes, and low for 4 x 4 process. It may not be appropriate to force the biggest log-modulus to $2n$. The independent design based on the SISO IMC-PID tuning also shows a sluggish response due to large integral times given by the SISO design for some processes. Simulations for the sequential autotuning method show its idiosyncrasies. That is. loops tuned first show very fast responses but the latter ones are slow. In contrast, the proposed method shows good responses that are well balanced.

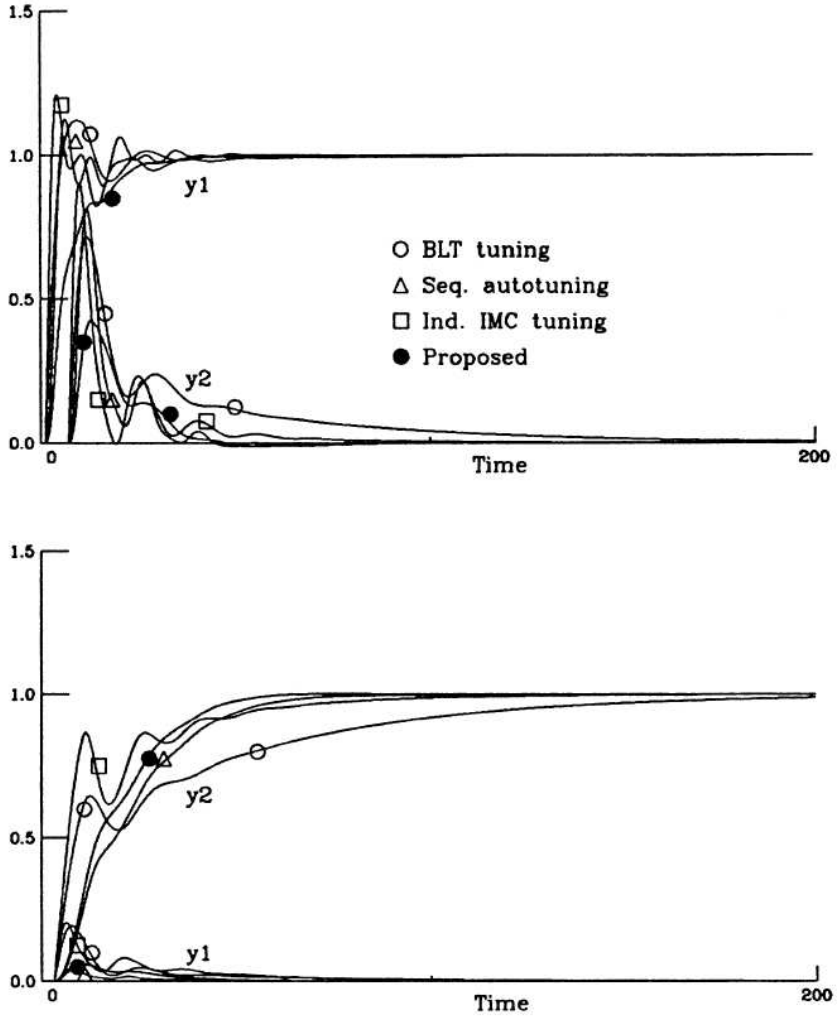


Figure 7.13: Set point responses for the WB column

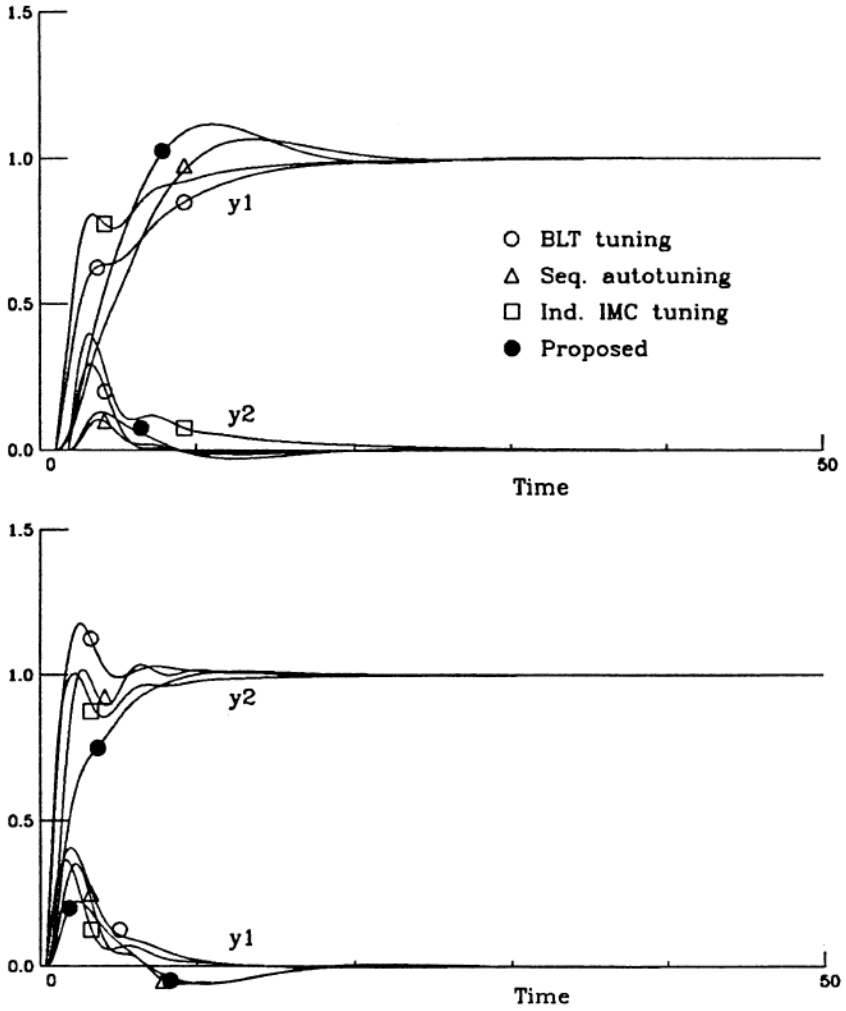


Figure 7.14: Set point responses for the VL column.

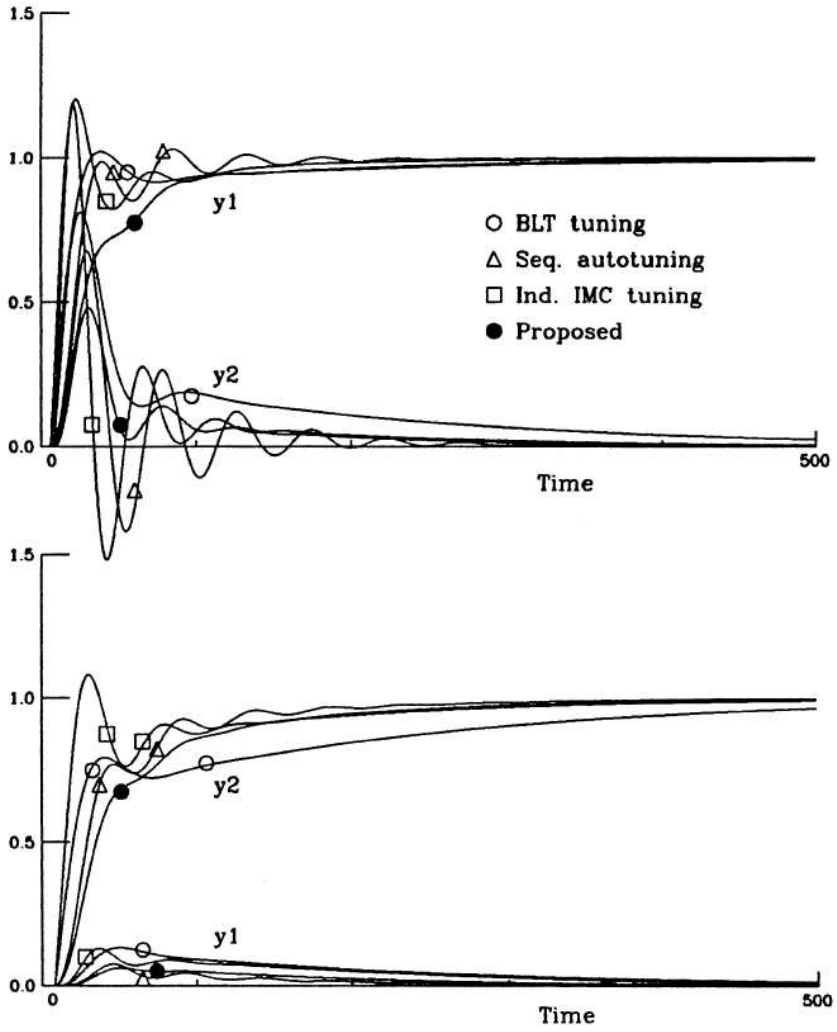


Figure 7.15: Set point responses for the WW column

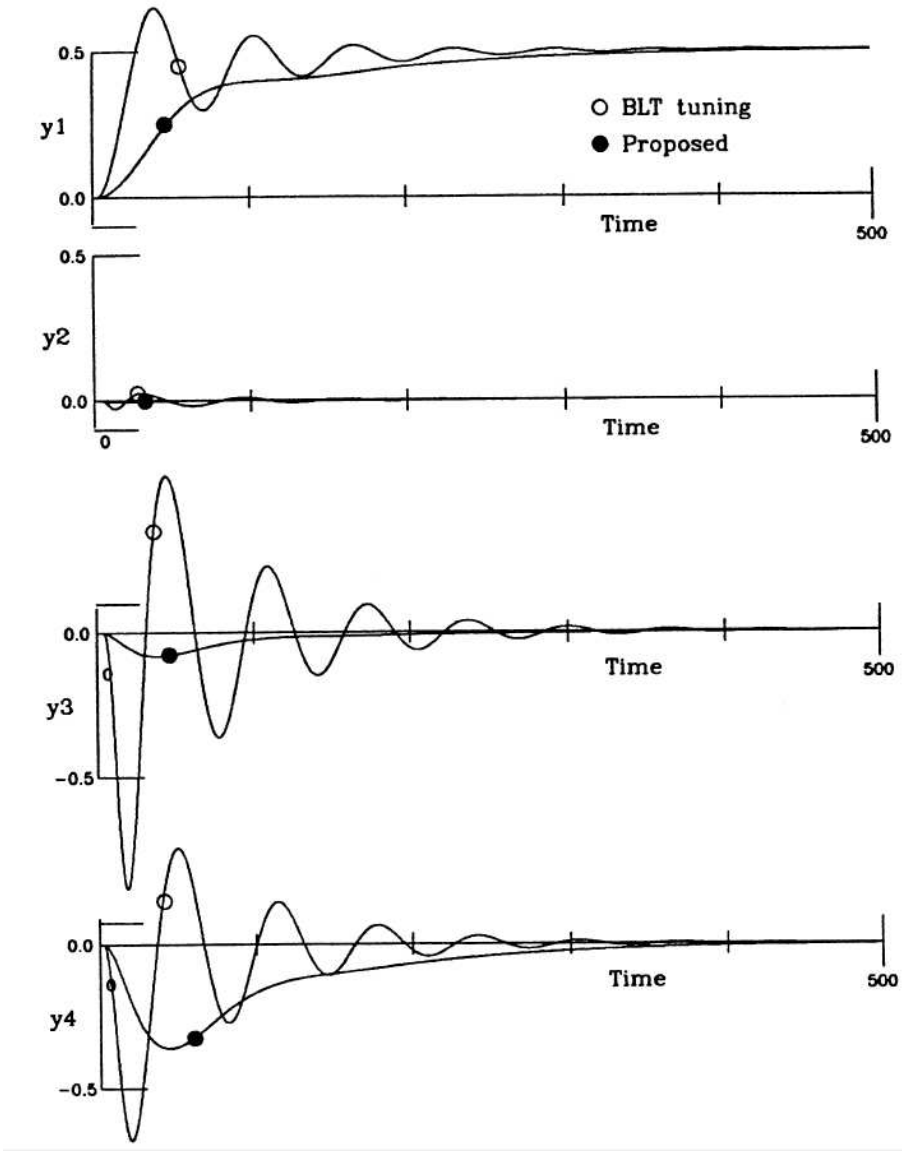


Figure 7.16: Set point responses for the A1 column

Table 7.1: Tuning results for various distillation column examples

		BLT Tuning	Sequential autotuning	Independent IMC Tuning	Proposed
WB	K_c	0.375	0.868	0.737	0.850
		-0.075	-0.0868	-0.103	-0.0885
	τ_i	8.29	3.246	17.2	7.21
		23.6	10.4	15.9	8.86
	$\sigma_{min}[I + (GC)^{-1}]$	0.61	0.33	0.47	0.42
VL	K_c	-1.07	-1.353	-1.54	-1.31
		1.97	3.36	2.37	3.97
	τ_i	7.1	3	7.5	2.26
		2.58	1.33	9.38	2.42
	$\sigma_{min}[I + (GC)^{-1}]$	0.72	0.40	0.70	0.53
WW	K_c	27.4	48.1	51.5	53.8
		-13.3	-25.4	-21.3	-20.3
	τ_i	41.4	18.99	63	31.1
		52.9	26.3	39	29.7
	$\sigma_{min}[I + (GC)^{-1}]$	0.63	0.17	0.33	0.26
A1	K_c	2.28			0.385
		2.94			6.190
	1.18				2.836
	2.02				0.732
	τ_i	72.2			34.72
		7.48			21.80
		7.39			19.22
		27.8			36.93
	$\sigma_{min}[I + (GC)^{-1}]$	0.073			0.18

Chapter 8

Application of Iterative Learning Control for Optimal Recipe Setup for a RTP Chamber Wafer Temperature Uniformity Control

8.1 Introduction

A conventional iterative learning control algorithm updates the next controller output from the previous run as follows,

$$u_{k+1} = u_k + \Gamma \frac{d}{dt}(y_{sp} - y)$$

where u_{k+1} and u_k are next and previous controller output trajectory vectors, respectively. An improved general form of the iterative learning control adds feedback information to above updates as

$$u_{k+1} = u_k + \Gamma \frac{d}{dt}(y_{sp} - y) + \text{feedback control term}$$

where the feedback control scheme can be either a conventional PID or more advanced techniques such as LQG, Model Predictive Control, or QDMC. This Chapter introduces an iterative method of updating RTP recipe through an iterative learning control and iterative recipe data update.

While there are a number of published research works (Choi et al.[19], Yang et al.[105]) involving the iterative learning control of the RTP temperature control system for multivariable cases (multi lamp power inputs and multi temperature

outputs), some industrial RTP systems are still utilizing the SISO control scheme. This is probably because the existing RTP system is not originally designed to handle MIMO control scheme. Unless the equipment manufacturer provides the solution for MIMO scheme, RTP process engineers needs to utilize existing software and hardware for the control system. Also, for certain RTP systems, the condition number of the process gain matrix can be large. This can cause some difficulties in MIMO control due to high sensitivity of the control performance due to model uncertainty. MIMO iterative learning control might have same difficulties due to the following factors:

- Perturbation in initial states
- Controller output drift due to system drift
- Control convergence is slow
- Control convergence is oscillatory

Even in the SISO case, it may be difficult to adapt the control hardware and software to build an iterative control scheme by modifying the current setup. This is mainly because the software source code is not available to the process engineers. However, it may still be possible to improve the control performance of certain single loop controlled RTP system.

8.2 Recipe Setup by Iterative Parameter Update

A schematic for lamp zones for a commercial or research purpose RTP chambers is shown in Figure 8.1. For a commercial RTP chamber such as AG Heatpulse 8108, different recipes for various products and processes are set up for the operation of each batch run. Key parameters defined in these recipes define are the lamp zone power ratios. Suppose that we are setting up a recipe for the RTP system used in this research. The three concentric lamp zones and temperature control pairs are shown in Figure 8.2. Typical lamp ratio values for recipe for each product and process are shown in Figure 8.3.

8.3 Algorithm

Once the standard recipe is set up by a service engineer from the manufacturer, there is no proper way for process engineer or the tool owner to tune the lamp power ratio. This Chapter suggests a systematic method for the determination of the power ratios. In summary, different from the iterative learning control with multiloop feedback control scheme, this method updates the lamp power ratios through iterative control with PI control feedback and iteratively calculates the power ratio of the other lamp zones to update the optimal ratio for the recipe.

When the second lamp zone and T_{c2} are paired and the controller output u_2 is updated as

$$u_{2,k+1} = u_{2,k} + \Gamma \frac{d}{dt}(y_{sp} - y_2) + \text{PID feedback} \tag{8.1}$$

Then u_1 and u_2 are updated using power ratios which are iteratively updated

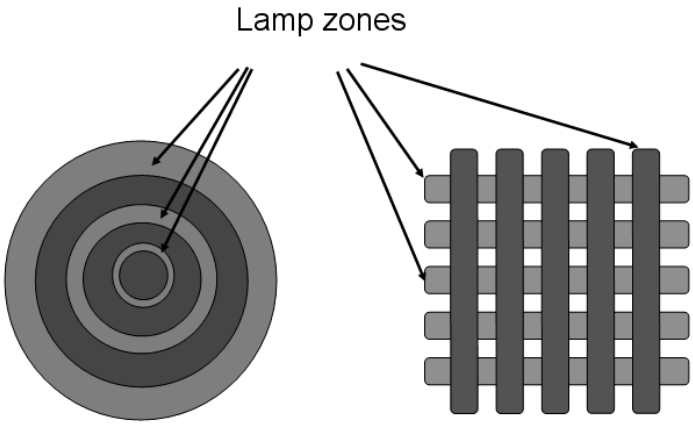


Figure 8.1: Lamp zones of the RTP chamber.

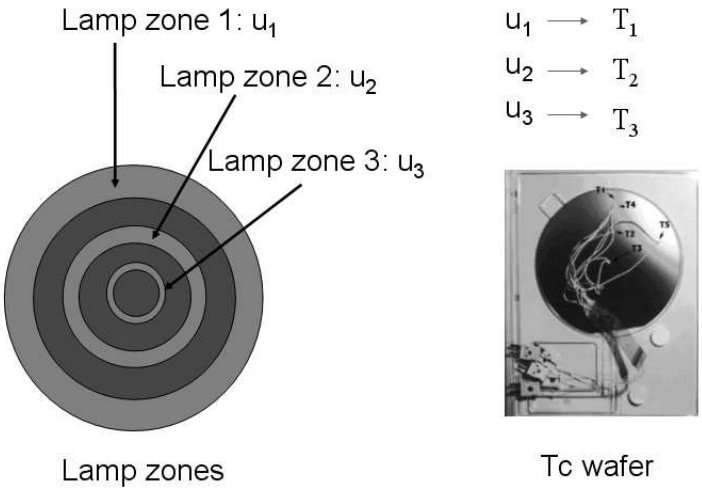
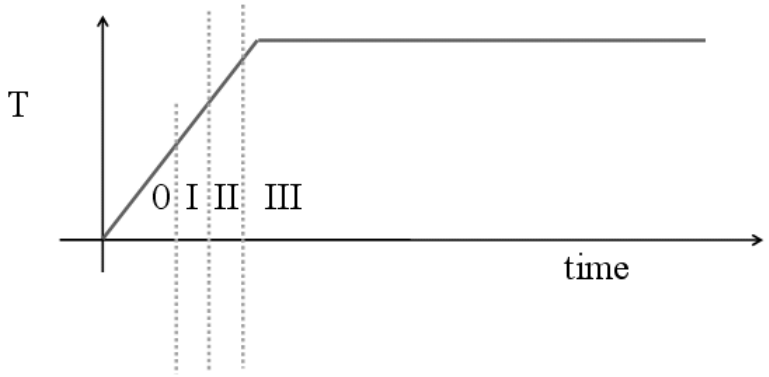


Figure 8.2: Lamp zones and thermocouple points for control pairing.



	Seg I	Seg II	Seg III
Zone I	30%	43%	30%
Zone II	20%	24%	20%
Zone III	40%	33%	16%

← Lamp power %

Figure 8.3: Power ratio values of the recipe

for each operating segments

$$u_{1,k+1} = \overbrace{a_{1,k+1}u_{2,k+1}}^{\text{SegmentI}} + \overbrace{a_{2,k+1}u_{2,k+1}}^{\text{SegmentII}} + \overbrace{a_{3,k+1}u_{2,k+1}}^{\text{SegmentIII}} \tag{8.2}$$

$$u_{3,k+1} = \overbrace{c_{1,k+1}u_{2,k+1}}^{\text{SegmentI}} + \overbrace{c_{2,k+1}u_{2,k+1}}^{\text{SegmentII}} + \overbrace{c_{3,k+1}u_{2,k+1}}^{\text{SegmentIII}} \tag{8.3}$$

$$\tag{8.4}$$

where for each segment the parameters are updated as

$$\begin{aligned} \text{Segment I}: \quad a_{1,k+1} &= a_{1,k} - \alpha_1 \text{MAX}(y_1 - y_s p) \\ \text{Segment II}: \quad a_{2,k+1} &= a_{2,k} - \alpha_2 \text{MAX}(y_1 - y_s p) \\ \text{Segment III}: \quad a_{3,k+1} &= a_{3,k} - \alpha_3 \text{MAX}(y_1 - y_s p) \end{aligned} \tag{8.5}$$

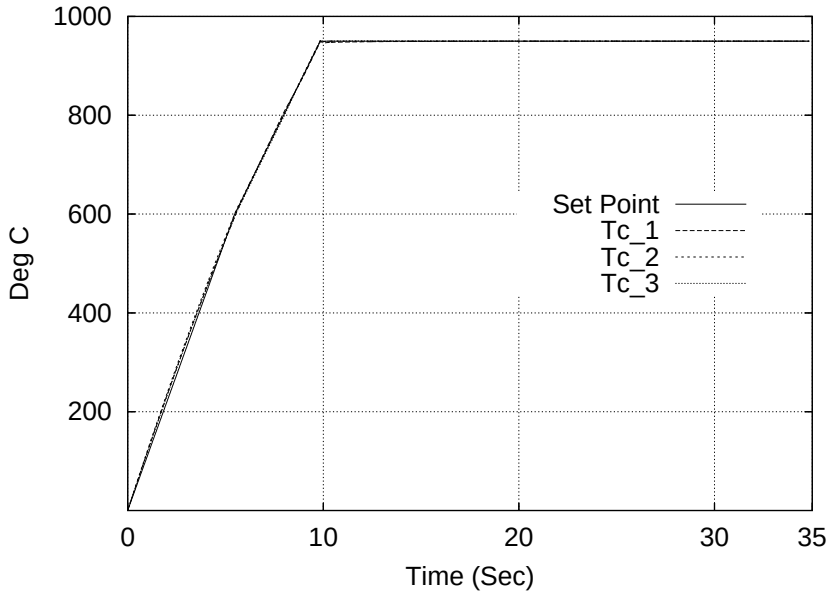
The same update algorithm applies to parameters c_1, c_2 and c_3.

8.4 Simulation Results

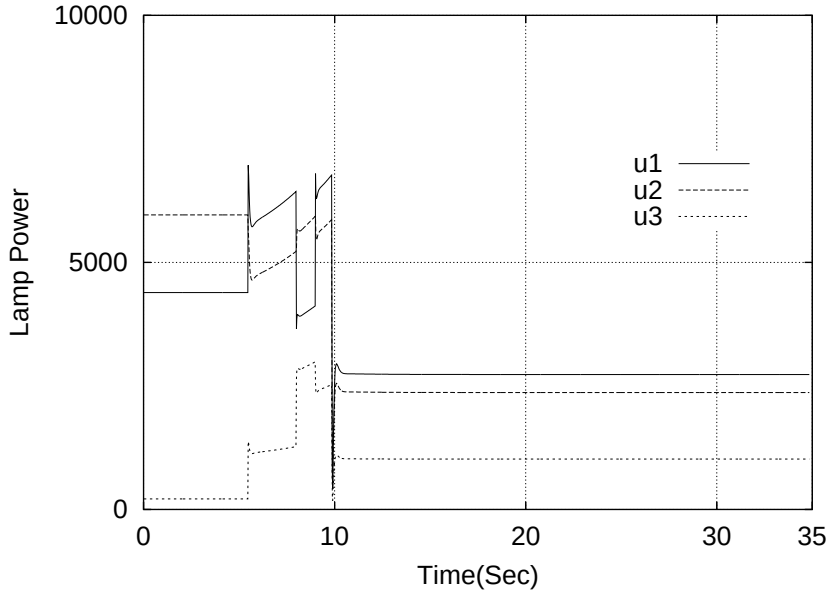
Using the model developed in Chapter 3 and the identified model from Chapter 4, a MATLAB program was developed to iteratively update the recipe parameters. Figure 8.4 shows simulation results after 40 iterations. The detailed temperature responses are shown in Figure 8.5.

Sum of the squared errors(SQE) between y_2 and y_{sp} and standard deviation (STD) over three temperature points as the iteration progresses are shown in Figure 8.6.

Squared error and parameter values for a and c parameters are plotted in Figure 8.7.



(a)



(b)

Figure 8.4: (a) Temperature response and (b) controller output levels after 40th iteration

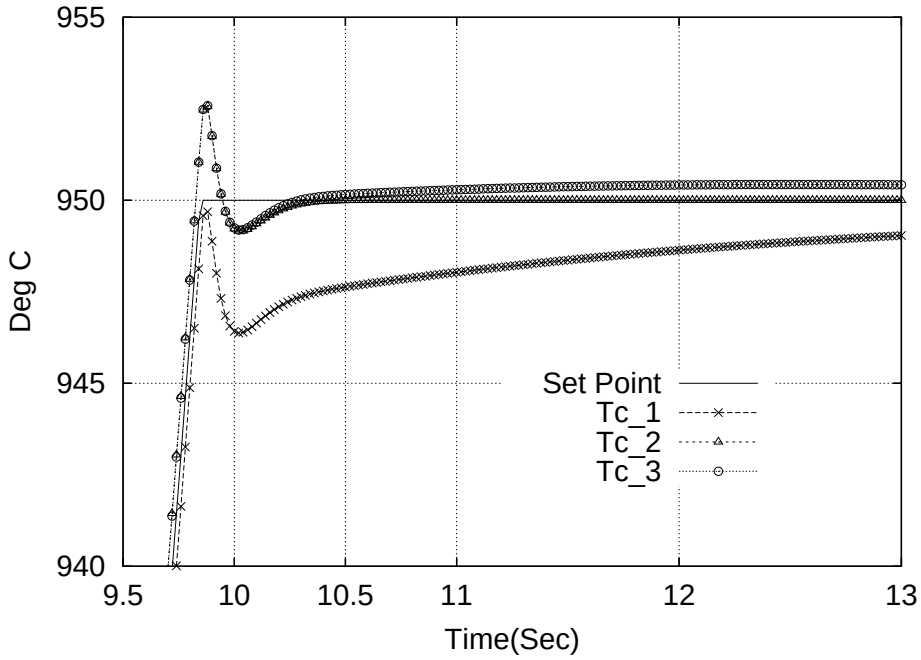


Figure 8.5: Detailed plot of the temperature response around 950°C

8.5 Conclusions

As shown in Figure 8.4 (a), overall temperature response for ramp and hold set point steps are satisfactory, detailed response curves in Figure 8.5 shows relatively closer set point tracking for T_{c2}, and about 3 °C and 1 °C deviation for T_c1 at $T = 10$ and $T = 13$ seconds, respectively. Both T_{c1} and T_{c3} follows the response of T_{c2}. This is expected because the lamp powers for u_1 and u_3 are the ratio of u_2.

Sum of squared errors(SQE) decreased exponentially at the beginning of iteration and slowly decreased until around the 30^{th} iteration, after that SQE increased

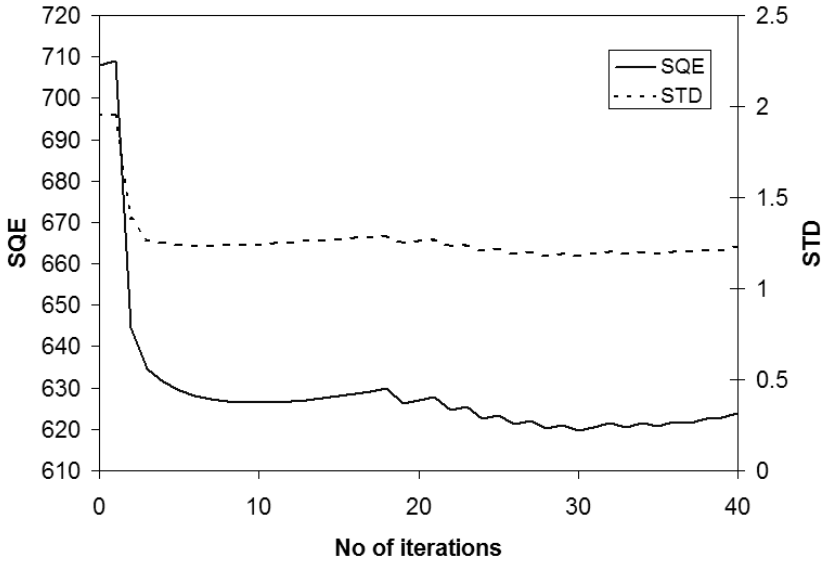


Figure 8.6: Sum of squared error and standard deviation as iteration progresses

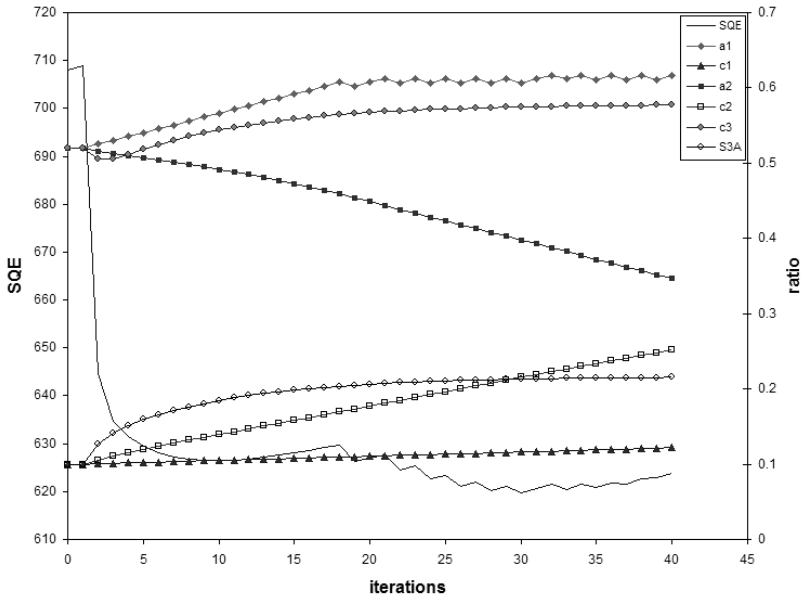
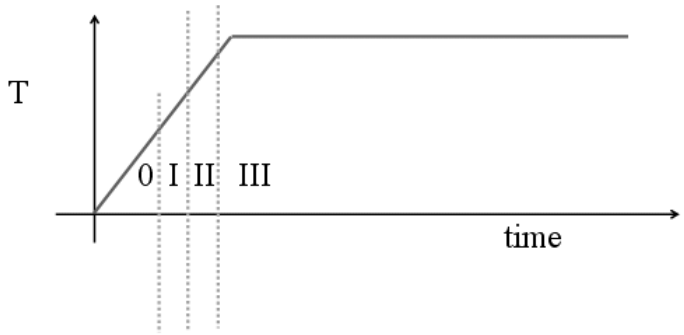


Figure 8.7: Sum of squared error and parameter values as iteration progresses



	Seg I	Seg II	Seg III
Zone I	61.6	34.7	57.7
Zone II	50.0	50.0	50.0
Zone III	12.2	25.2	21.5

Lamp power %

Reference control input

Figure 8.8: Final recipe values for each segments

slowly. The standard deviations of the three temperatures also followed SQE as the iteration progressed. So, for this case, the iteration steps beyond the 30^{th} iteration are not necessary. Parameters for power ratio for segment I and II are converging to final values, but those for segment II (a_2 and c_2) continuously decreased (a_2) and increased (c_2), without oscillations. Since output variables for these parameters are tracking the target set points, it would not be a serious problem. It should be understood that lamp zones I and III are compensating the power distribution to control wafer temperature for this specific case.

Recipe parameter values are determined from the 40^{th} iteration and are shown in Figure 8.8.

Chapter 9

Conclusions and Recommendations

9.1 Conclusions

In order to understand and control wafer temperature in RTP, a RTP chamber was set up for testing. As described in Chapter 2, a three input (lamp powers), three output (wafer zone temperature) computer control system was set up for testing. The control system can perform multiloop PID control, gain scheduling PID control and multivariable control.

In Chapter 3, a heat balance based mathematical model was developed for RTP temperature control and identification. Process gain K_{ij} and dominant time constants τ_i can be obtained from experiments. These parameters can be analytically derived from the heat balance equation and are functions of effective view factor F_{ij}, effective emissivity D_i, temperature and convective heat transfer coefficient. Since F_{ij} and D_{ij} are also dependent on the wafer total emissivity A_ϵ, wafer zone to zone view factor V, and lamp to wafer view factor W, a parameter estimation scheme was derived. Consequently sensitivity of process gain to temperature $\partial K_{ij}/\partial T_i$ and time constant to temperature $\partial \tau_i/\partial T_i$ of RTP chamber can be explicitly represented from a set of identification experiments.

In Chapter 4, closed-loop identification experimental results are presented. Closed-loop identification methods used are explained in detail. Experimental runs

ranging from 300°C to 700°C set points were performed and identification results for key process parameters, K_{ij} and τ_i, are presented. The first order plus time delay model appears to be adequate for fitting the data.

In Chapter 5, iterative identification methods are proposed for ill-conditioned processes such as RTP that are very sensitive for the element uncertainties. For such processes, element uncertainties can be amplified and classical element-by-element identification cannot be used. Methods based on SVD, QR, and LU decompositions that diagonalize and triangularize the process iteratively were investigated. Simulations have shown that the SVD method has the largest convergence ranges of identification errors and other methods, including the Koung and MacGregor method [45], have similar convergence ranges. The SVD, QR, and KM methods are all based on transformations with orthogonal matrices and hence can not easily be extended to obtain dynamic process transfer functions. On the other hand, the LU method uses analytical transformations and can be applied to dynamic processes.

In Chapter 6, new measures of some practical uncertainty effects on the closed-loop control performances, stability, and model inversion were developed, which build on previous work on decentralized control by Skogestad and Morari[85] and Chen et al.[13]. The proposed simpler norm bounds (the uncertainty condition numbers) are induced norms of matrices that can be constructed very easily, like the RGA. In addition to the diagonal input uncertainties, this analysis treats non diagonal uncertainties and elementwise model uncertainties with minor modifications, which are useful results for multiloop control.

In Chapter 7, various multiloop PID tuning methods are compared and a

simple tuning method is proposed. Currently used multiloop controller tuning methods include; 1) Detuning method, 2) Sequential closing method and 3) Independent design method. In this Chapter, a simple multiloop controller tuning method is presented that is based on utilizing trial and error tuning method for an SISO process. Examples are shown for comparison of the various tuning methods. RTP chamber lamp power/wafer temperature multiloop control problem is also simulated for the tuning method.

In Chapter 8, a simple SISO based iterative technique is developed for the multivariable RTP system. An optimal recipe generation is possible through iterative update of lamp power ratio at each iterative run. This technique can be used for industrial RTP chamber temperature control applications where SISO control is used but multiple lamp power adjustments and multiple wafer temperature measurements are available.

9.2 Recommendations

- Wafer emissivity deviation, assuming correctly calibrated pyrometer temperature measurement, can occur due to unwanted deposition on the measurement side of wafer. This can be detected from closed-loop identification of the process gain and time constant. While it is still necessary to verify the sensitivity of closed-loop identification from IR pyrometer measured temperature, it is strongly recommended to prepare test wafers with various backside depositions(as close to material properties as expected from FAB environment) and perform an extensive study on measurement error due to IR pyrometry. Since

this test should be performed on-line, closed-loop identification is preferred to open-loop identification.

- The IR pyrometer is not able to detect radiation signal from the wafer under certain temperatures(500°C to 600°C). In practice, the wafer is heated open-loop until the IR pyrometer senses the signal from the wafer and the control loop can start feedback control. In this time interval, the temperature control loop is not closed. Also, due to the possible deviation from nominal wafer backside emissivity, controller output(manipulated variable) level can deviate at steady state. A study on providing upper and lower bounds of controller output (and monitoring it) at each stage of RTP temperature control would be useful.

- A temperature measurement technique needs to be improved; the current IR pyrometer measures radiations from the wafer as well as from the quartz window and reflections of lamp radiation. The quartz plate can be made thinner. Lamp radiation can be included in pyrometer detection (model based control) and subtracted a priori.

- A temperature sensor to measure quartz plate temperature should be installed. This is mainly for testing purposes to be included in pyrometer measurement compensation.

- Inert gas purging is recommended for the wafer. This can reduce the temperature drift effect due to the quartz plate and shorten the wafer cooling time.

- For plasma CVD experiment, a larger diameter high density plasma supply mechanism based on a helical coil is recommended. Distance from wafer to the plasma source should be minimized.

Appendices

Appendix A

Hardware Specifications and Wiring

A.1 Process Control Hardware and Software

The acronym SCADA stands for *Supervisory Control And Data Acquisition.* Supervisory control which is expressed as the on-line calculation of set points(page 474 of [78]) configuration consists of a digital computer and digital or analog front end controllers which control the process directly. A process (or a plant) is a complex, interacting entity, and the optimum operating strategy can only be ascertained after considering the combined effect of many different options and constraints, such as[78][89]:

- Cost of raw materials and utilities
- Storage and warehousing capacities
- Value of the Products
- Composition of raw materials and products
- Constraints on the operation(temperatures and pressures must be within certain limits)
- Specifications on the products(allowed levels of impurities etc.)

Namely, the set points of the individual unit processes should be set on-line from the digital computer which analyzes the process data and calculates the set points through a certain optimization routine with the above options and constraints.

The acronym DCS stands for *Distributed Control System*. In DCS, the bottom level analog controllers are replaced with more flexible controllers on a one-to-one functional basis, and these digital controllers are connected through some form of communication network. Whether or not a high-level digital computer is present for data analysis and/or more sophisticated control, the control system can be called a distributed control system because of the network approach[78].

Presently SCADA or DCS can be equally used to describe a digital computer control system due to the remarkable development of the computer industry. Namely, the cost of the computer which can function as a supervisory computer has become inexpensive enough that most process control systems utilize some kind of computer(normally workstation level computer) for higher level computation and some type of data communication links connect individual bottom level controllers and a higher level digital computer can be easily included to the network.

In Figure A.1, the general schematic of the control system hierarchy for this research is introduced. The TI 545 PLC performs data acquisition and simple(can perform relatively complex routines) control tasks; the Windows NT workstation is used for higher level system identification and control purposes; the 386/33 SCO Unix workstation is used for a graphical interface; and SUN/IPX workstation is used for off-line data analysis and word processing.

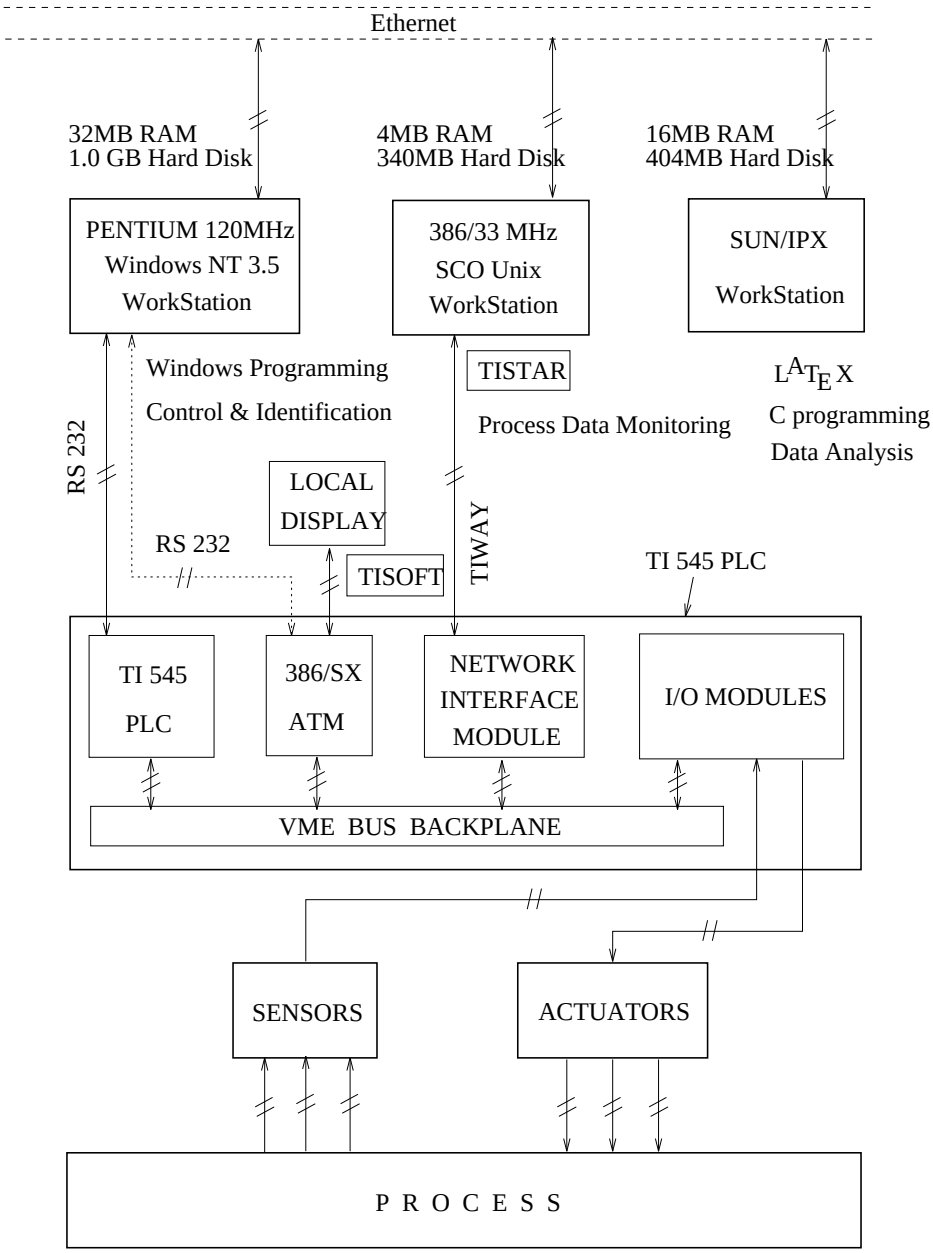


Figure A.1: Overall control system schematic

Process data is interfaced to PLC through various I/O moules located in the TI 545 PLC rack, and the process data are read or written through the VME bus backplane locate at the back of PLC rack. 545 PLC itself performs various tasks including PID loops whose set points and tuning parameters can be set by the NT workstation through serial communication(RS 232 C). Process data can be sent to 386/33 workstation and operator can send process operating instructions to the process through the TIWAY data highway which links the network interface module in the 545 PLC and the SCO unix workstation.

This system can be called a DCS, because the TI 545 PLC can have the remote bases as explained in the next section and some actuators have their own PID controllers implemented. Also, obviously we can classify the system as a SCADA system because of the presense of the Windows NT workstation or 386/33 SCO unix workstation which performs very high level computation.

A.1.1 TI 545 PLC

The TI 545 PLC provides PID loop control, sequential control through relay ladder logic(RLL) programming, some higher level programming capability through special function programming, and communication to other computer through RS 232 or RS 422 ports. It also reads and writes data from and to each input/output module. Detailed descriptions are written below, and for further information refer to [97].

For this research, 545 PLC is housed in a Model 505-6516 base assembly, which may consist of as many as 16 I/O modules. I/O modules(see Figure A.5)

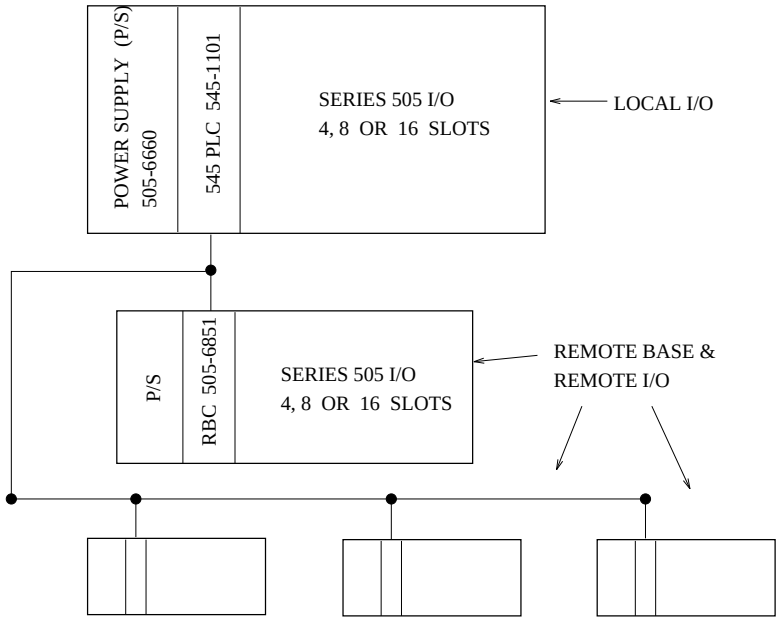


Figure A.2: System components for PLC Model 545

used are Thermocouple Input Module(505-7028), 24V DC Input(505-7028), 24V DC Output(505-4732), 8 In/ 4 Out Analog(505-7012), High Speed Counter(505-7002), Relay Output(505-4916), Input Simulator(505-6010), Output Simulator(505-6011), and TIWAY I Network interface(505-7339).

TI provides 4, 8 and 16 slots base assembly, and up to 15 additional base assemblies can be connected through remote base controllers to a 545 PLC. The base assembly which contains 545 PLC is numbered 0, and I/O modules installed in this base is called local I/O. I/O modules in other bases are called remote I/O and the base is called remote base. Each base comprises up to 1024 I/O points. So a 545 PLC with 16 slots connected to 15 remote bases with 16 slots has total of 256

slots, and can handle up to 16384 I/O points(refer to Figure A.2). In this research only one base assembly is used.

The model 545 PLC scan sequence is shown in Figure A.3. Each scan step is explained below verbatim from the manual[97].

I/O Update. During the I/O cycle update the PLC stores data obtained from the I/O bases into the image registers. The length of the I/O update cycle is dependent upon the number of bases and types of modules. All I/O points are fully updated each scan.

Special Function Module Communication. Special Function(SF) module communication is divided into two time slices. In the first slice, the following actions take place.

- Deferred task codes from a previous scan for which processing has been completed are transmitted to the SF modules.
- Remote base communication ports are polled for service requests.

During the second time slice, following RLL execution, the controller processes service requests from SF modules and remote base communication ports.

Each SF module that requires service adds scan time amount of which is dependent on the type of module and task. Each type of module is allowed a certain number of task code requests, block transfers, or store-and-forward operations per scan. Once these are completed, this function is terminated. Some task codes can

be deferred, and these are processed by the Deferred Task Code Handler during the analog task time slice described below.

Ladder Logic Cycle. The PLC executes the RLL program. The RLL runs to completion but can be momentarily interrupted by certain functions such as watchdog timers and base *wake-up* transmissions.

Local Communication Port Service. The PLC processes tasks received through the local communication ports, which are located on the PLC itself. Read/Write and other non-deferred task codes that have been received through these ports are run to completion.

Analog Tasks. The analog portion of the scan comprises five general types of time slices.

- Loop processing – loop operations are executed.
- Analog alarm processing – analog alarm operations are executed.
- Special Function program processing – time slices are provided for cyclic programs, priority programs, non-priority programs, and Special Function subroutines. Restricted programs, which are called by loops or analog alarms, are processed during the loop or analog alarm time slice.
- Deferred Task Code processing – deferred task codes are processed.
- Diagnostic processing – run-time diagnostics.

Analog tasks are also processed during windows of halted activity that occur during other time slices. RLL execution is not interrupted by analog tasks. Analog

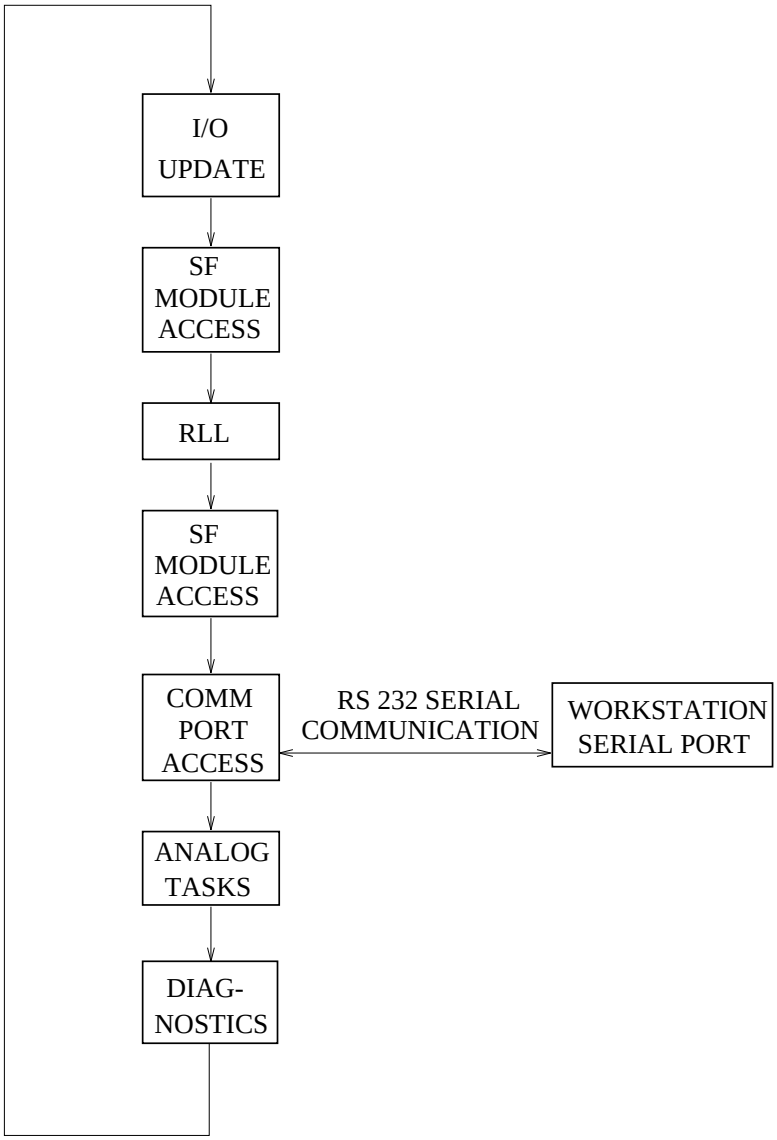


Figure A.3: Scan components for PLC Model 545

31	30 23	22 0
SIGN	BIASED EXPONENT	FRACTIONAL PART
1 BIT	8 BITS	23 BITS

Figure A.4: Format of a real number

tasks are guaranteed execution once per scan provided there is processing to be done.

A.1.2 Data storage and I/O configuration

Real numbers: Real numbers are stored in the single-precision 32-bit(two words) binary format(IEEE 754 representation[56]). Bit 31 is the *sign*, bits 30 through 23 are the *biased exponent*, and bits 22 through bit 0 are the *fraction*. That is, the exponents field is 8 bits wide and the fraction field is 23 bits wide, as shown in Figure A.4. The *sign* is 0 for a positive number, 1 for a negative number.

The *biased exponet* is the number's *binary exponent* plus a *bias* of 127. The bias guarantees that all exponents are positive, simplifying the probelm of representing negative exponents. The binary exponent may have any value between -126 and +127($1 \geq$ biased exponent ≥ 254). The values -127 and +128 are reserved for exception handling.

The *fraction* represents the fractional part of the mantissa of a floating-point number. The integer part of this mantissa is always one and is not represented.

Thus the value of any single-precision floating-point number is therfore:

$$(-1)^{**}\text{sign} * (1 + \text{fraction}) * 2^{**}(\text{biased}_{\text{e}}\text{xponent} - 127) \tag{A.1}$$

Detail data formats for *signed integers*,*unsigned integers*, and *double integers* can be found in [97].

The memory types or variable types used in PLC programming are various. There are WX, WY, V-Memory, S-memory and T-Memory which were used extensively in this research. There also exist L-Memory for RLL instructions, Drum Memory, Shift Register Memory, Table Move Memory, One Shot Memory, Status Word memory, and Constant Memory(K-Memory), which were not used in this research.

WX, WY (Word Image Register): The word image register is an area of memory in PLC that is reserved to hold all 16-bit word inputs(WX) and outputs(WY). As explained earlier in PLC scanning description, the word image register is updated prior to evey PLC scan. Because the word inputs and outputs share one image register, the same reference number can not be assigned to both WX and a WY. In model 545 PLC, 1024 word I/O points are supported.

For example, in this research as shown in Figure A.5, 5 of the thermocouple temperature readings are assigned as WX9, through WX13. Also, WX81 and WX82 are assigned to pyrometer 1 and 2, respectively. Output values, lamp controller output levels, are written to WY93, WY94, and WY95. WX and WY values are integer values ranging from 0 to 32,000.

X, Y (Discrete Image Register): An area of memory within the PLC called the discrete image register is reserved for maintaining the status of all discrete inputs(X) and and outputs(Y). The discrete image register is updated prior to every PLC scan when the status of the discrete I/O points is read and written.

V-Memory: A block of memory within the PLC allocated for user operation is called Variable Mermory(V-Memory). It can be used to store math operation results, to check the status of a bit within a word by holding the word in V-Memory, and you can write data values directly to V-Memory.

For example, for K-type thermocouple input WX9, when SCALE option is selected, WX9 value can vary from 0 to 32,000. Maximum and minimum ranges for K-type thermocouple is 1320°C and -220°C , respectively. Then if you want the temperature expressed in °C stored in V-Memory, V91,

$$V91. = \frac{WX9 - 0}{32000 - 0} \times \{1320 - (-220)\} \tag{A.2}$$

S-Memory: A block of memory within the PLC allocated for loops(PID), analog alarms, and Special Function programs is called Special Memory(S-Memory).

T-Memory: A block of memory within the PLC allocated during run time whenever a Special Function progaram is run. This memory type is 16 words long and is called Temporary Memory(T-Memory) and it is not saved when the program has completed running. The PLC writes data related to the Special Function program to the first seven words.

The I/O modules are installed as shown in Figure A.5 and the I/O module

definition chart from TISOFT(see section **??**) is shown in Figure A.6. The first column of the I/O definition chart shows the slot numbers in which the I/O modules are installed. Each I/O module occupies from 1 to 3 slots in the base. In Figure A.5, the slot numbers which each I/O module is installed are typed in the parenthesis.

386/ATM module[100] is the first I/O module installed(the Power Supply and the Model 545 PLC is not considered as I/O modules) and occupies 3 slots(01 to 03 slots). Namely, 386/ATM module communicates with PLC using 4 WX and 4 WY word image registers. Each image register's I/O address becomes WX1, WX2, WX3, WX4, WY5, WY6, WY7, and WY8.

Thus the next module, Thermocouple Input moduleti:tc, starts with an I/O address 09(after WY8 of 386/ATM module's). And since this module has 8 thermocouple input ports and since it reads from the process, the word image registers are WX9 through WX16(8 image registers).

The 24VDC Input module[92](I/O address 5) uses 32 single bit input discrete image registers(X), and the X image registers' addresses are X17 through X48. The 24VDC Ouput moduleti:dout(I/O address 6, 7 and occupies 2 slots), uses 32 single bit output discrete image registers(Y), and the addreses are X49 through X80. For DC Output module, external DC power supplies should provide necessary power to be used to drive relays or coils.

There are four 8 In/4 Out Analog modules[99](I/O addresses 8, 9, 10 and 11). And each module uses 12 WX image registers for the analog input and 4 WY image registers. As for DC Output module, each analog output points need external DC power supplies. And the image register addresses are WX81 through WX92 for

analog input, and WY93 through WY96 for analog output signals for the 1st 8 In/4 Out Analog module(slot 8). Other three following Analog modules' (in slots 9, 10 and 11) image register addresses are assigned in the same way.

The High Speed Counter module[96] in slot 12 uses WX145, WX146 and WX147 input image registers, and WY148 through WY152 output image registers. The Relay Ouput module[94] in slot 13 use 16 single bit discrete output registers(Y153 through Y168). The Input Simulator module[93] in slot 14 uses 32 single bit discrete input image registers X169 through X200 and the Output Simulator module[93] in slot 15 uses 32 single bit discrete output image registers Y201 through Y232.

Finally, the TIWAY I Network Interface module[95] in slot 16 uses 8 single bit discrete output registers(Y233 through Y240).

The actual memory addresses used in this research is listed in Table A.1.

A.1.3 PLC programming

The lowest level PLC programming tools used in the TI 545 PLC is relay ladder logic(RLL). RLL can be programmed by TISOFT. TI's product Application Productivity Tool(APT) is a higher level programming language which is user friendly and can be compiled into RLL to be implemented for process control. The TISOFT also provides LOOP, which is a built in PID controller and SFPGM(special function program), which helps LOOP or RLL programming. In this research, LOOP and SFPGM are mostly used for PLC programming. The LOOP programming will be explained first followed by the SFPGM programming.

Table A.1: PLC memory address table

Variable description	Loop 1	Loop 2	Loop 3	Loop 4	Loop 5	remark
Measured variable	TC-1 WX9	TC-2 WX11	TC-3 WX13	Pyro-1 WX81	Pyro-2 WX82	integer
Controller output	WY93	WY94	WY95	WY93	WY95	integer
VFLAG (LVF)	V121	V131	V141	V151	V161	integer
Check bit	C11,12,13	C31,32,33	C51,52,53	C71,72,73	C91,92,93	bit
LSP	V11	V21	V31	V41	V51	
LKC	V13	V23	V33	V43	V53	
LTI	V15	V25	V35	V45	V55	
LTD	V17	V27	V37	V47	V57	
LTS	V19	V29	V39	V49	V59	
LMN	V123	V133	V143	V153	V163	
LMX	V125	V135	V145	V155	V165	
Gain scheduling LKC/LTI	V211/231	V251/271	V291/311	V331/351	V371/391	range 0
	V213/233	V253/273	V293/313	V333/353	V373/393	range 1
	V215/235	V255/275	V295/315	V335/355	V375/395	range 2
	V217/237	V257/277	V297/317	V337/357	V377/397	range 3
	V219/239	V259/279	V299/319	V339/359	V379/399	range 4
	V221/241	V261/281	V301/321	V341/361	V381/401	range 5
	V223/243	V263/283	V303/323	V343/363	V383/403	range 6

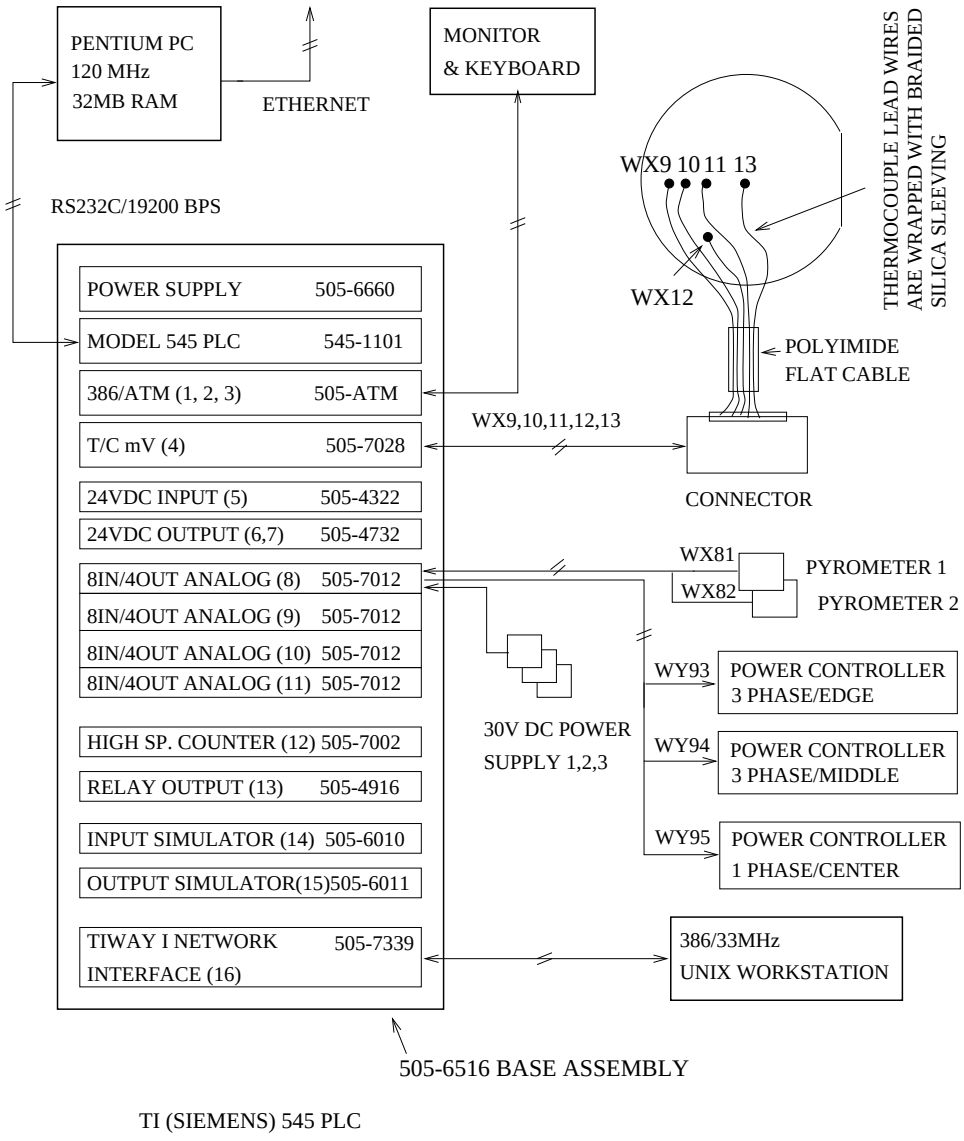


Figure A.5: Wiring details of the RT/RPCVD control system

```
confio.txt        Thu Jul 11 15:15:27 1996        1


    TISOFT
 DATE: 00-00-00     UT CHEM. ENG. CPE 5.420 545 PLC CONFIG.  545  LOOPALL   RN

              I/O MODULE DEFINITION FOR CHANNEL ... 1   BASE ..... 00

                          I/O       NUMBER OF BIT AND WORD I/O    SPECIAL
   BASE        SLOT    ADDRESS         X     Y    WX    WY      FUNCTION
  ENABLED
                01 .... 0001 ........ 00 .. 00 .. 04 .. 04 ....... YES
                02 .... 0000 ........ 00 .. 00 .. 00 .. 00 ....... NO
                03 .... 0000 ........ 00 .. 00 .. 00 .. 00 ....... NO
                04 .... 0009 ........ 00 .. 00 .. 08 .. 00 ....... NO
                05 .... 0017 ........ 32 .. 00 .. 00 .. 00 ....... NO
                06 .... 0049 ........ 00 .. 32 .. 00 .. 00 ....... NO
                07 .... 0000 ........ 00 .. 00 .. 00 .. 00 ....... NO
                08 .... 0081 ........ 00 .. 00 .. 12 .. 04 ....... NO
                09 .... 0097 ........ 00 .. 00 .. 12 .. 04 ....... NO
                10 .... 0113 ........ 00 .. 00 .. 12 .. 04 ....... NO
                11 .... 0129 ........ 00 .. 00 .. 12 .. 04 ....... NO
                12 .... 0145 ........ 00 .. 00 .. 03 .. 05 ....... NO
                13 .... 0153 ........ 00 .. 16 .. 00 .. 00 ....... NO
                14 .... 0169 ........ 32 .. 00 .. 00 .. 00 ....... NO
                15 .... 0201 ........ 00 .. 32 .. 00 .. 00 ....... NO
                16 .... 0233 ........ 00 .. 08 .. 00 .. 00 ....... YES
FROM PLC
                                                              545  LOOPALL   RN
 EXIT-F1 SHOW-F2 READDK-F3 WRITDK-F4 READPC-F5 WRITPC-F6 READBS-F7  CLRBS-F8 +
```

Figure A.6: I/O module definition chart for TI 545 PLC

A digital version of the PID controller can be expressed as either by the position algorithm or by the velocity algorithm[78]. The position algorithm is employed for this reserch and the controller output m_n at the nth sampling time is

$$\begin{aligned} m_n &= m_0 + K_c\Big[e_n + \frac{\Delta t}{\tau_I}\sum_{k=1}^{n} e_k + \frac{\tau_D}{\Delta t}(e_n - e_{n-1})\Big] \\ &= m_0 + K_c e_n + \frac{K_c \Delta t}{\tau_I}\sum_{k=1}^{n} e_k + \frac{K_c \tau_d}{\Delta t}(e_n - e_{n-1}). \end{aligned} \tag{A.3}$$

When $K_i = K_c\Delta t/\tau_I, K_r = K_c\tau_d/\Delta t$, and $e_n = S_p - P_{v,n}$, in which S_p is a set point and $P_{v,n}$ is a process variable at the nth sampling time, equation A.3

becomes(assuming that S_p is a constant)

$$m_n = m_0 + K_c e_n + K_i \sum_{k=1}^{n} e_k - K_r(P_{v,n} - P_{v,n-1}). \tag{A.4}$$

The PID position algorithm used in the TI 545 PLC calculates the bias($m_{x,n}$) and controller output(m_n) at the nth sampling time as

$$m_{x,n} = m_0 + K_i \sum_{k=1}^{n} e_k \tag{A.5}$$

$$m_n = m_{x,n} + K_c e_n - K_r(P_{v,n} - P_{v,n-1}). \tag{A.6}$$

As mentioned before, LOOP and SFPGM programming is done by using TISOFT on TI 545 PLC. Basically, user has to provide a set of data to PLC to build a loop. The PLC programmer should fill out the loop programming table shown in Figure A.7 with the loop data set such as shown in Table A.1.

Namely, the control system designer should set up a detail plan on how many loops and what I/O memory addresses would be used for the loops as shown in Table A.1. Frequently used loop variable names are listed in Table A.2. Refer to the Programming Reference Manual[98] for detail information.

Some important items in the loop programming table(Figure A.7) will be explained. Wheter the position or velocity algorithm would be used are selected by typing POS or VEL, respectively.

The V-Flags contain the operational data for a loop. The V-flags coorespond to individual bits making up the 16-bit word LVF(mnemonic). The first three V-Flags(bits) are designated as control flags. If one create a V-Flag table in V-memory, as in this research, the PLC reads these three bits in the LVF word. Bits 4-15 are

Table A.2: Loop variable names and types

Name	Mnemonic	Units	Real Only	Integer Only
Loop Setpoint	LSP	eu		
Loop Process Variable	LPV	eu		
Loop Gain	LKC	%/%	x	
Loop Reset	LTI	min	x	
Loop Rate	LTD	min	x	
Loop Sample Rate	LTS	sec	x	
Loop Output	LMN	%		
Loop Bias	LMX	%		

overwritten by PLC. Loop modes(Manual(bit 1 on), Auto(bit 2 on) and Cascade(bit 3 on)) can be set by setting 1 to 3 bits.

Writing "1000 0000 0000 0000(or 8000 in hexadecimal)" or "0100 0000 0000 0000(or 4000 in hexadecimal)" to V121 will set the loop 1 into *manual mode* or *auto mode*, respectively.

Sampleing rate(LTS), pocess variable(LPV) address, loop output(LMN) address, loop gain(LKC), reset(LTI) and rate(LTD) can be either typed into the loop programming table or set by the SFPGM. By attaching the loop number to the end of each variable name, each loop mnemonic varaible can be identified. Using the data provided in Table A.1, *LSP1 := V11* in SFPGM(figure A.8) assigns the set point value in V11 to LSP1(loop 1 setpoint).

SFPGM can perform a calculation on any constant, variable, or I/O point. SFPGM can be called either from LOOP or from RLL program. For the former case, one cah schedule the SFPGM call to be made when the process variable(PV),

setpoint(SP), or output is read by PLC from process.

If the SFPGM call is scheduled on PV, which is the same as the current setting in Figure A.7, and when the loop is in AUTO, CASCADE, or MANUAL mode, the SFPGM is executed every 2.0 seconds or at the sampling rate, whichever is less. In this way the alarms for the process variables can be activated even if the loop calcualtions are not being performed.

When SF program is called on SP, and when the loop is in AUTO or CASCADE mode, the SFPGM is called at the sample rate and T2=2[1] always. When the loop is in MANUAL mode, the SFPGM is not called for execution. Thus for the calculations which are essential for process monitoring even under MANUAL MODE loop operation, this is not a good option.

SF programs called on PV or SP are executed after PV or SP have been determined by the loop but before any processing has been performed based on the values obtained. This allows SF programs to manipulate the PV or SP before the loop uses them for output adjustments.

For the case when calculation is scheduled on output, and when a loop with a sample time of less than 2.0 seconds calls an SF program, the SF program is actually called twice for every loop calculation. Namely, following steps are carried out for this option.

- After PV and SP are determined, the SF program is called on SP(T2=2). This call allows for PV and SP manipulation before PV alarming and loop

[1]For explanation on T memory see Appendix

calculations are run. The loop calculation is then performed and the resultant output value is placed in LMN.

- The SF program is then called on output(T2=5) to allow for manipulation of the loop output value in LMN before this value is written to the loop output address.

In summary, for the process requiring fast sampling time(less than 0.5 seconds), and in case when the PV monitoring for the loops with MANUAL MODE is important, SF program call should be made on PV. Explanations on other items in the loop progamming table can be found in [98].

Programming the SFPGM is similar to programming BASIC language. The SFPGM programming language is more oriented to TI 545 PLC programming than normal all purpose BASIC langugae. A SF program(SFPGM) consists of a set of instructions that can be called from loops, alarms, or from the RLL program, much like a GOSUB subroutine in a BASIC program or a function in a C language program.

SFPGM can derive solutions for complex programs that would require extensive RLL programming and consume large blocks of ladder memory. Operations such as mathematical calculations, IF-THEN statements, unit and number format conversions, table transfers, data consolidations, etc., can be done with an SFPGM. Typically, these types of operations either cannot be done with the RLL instruction set or they involve complex RLL programming. Table A.3 lists alphabetically the staments used in SFPGM programs. And Figure A.8 shows a typical SF pro-

Table A.3: SFPGM Statements

Statement	Function
BCDBIN	Convert BCD(Binary Coded Decimal) To Binary
BINBCD	Convert Binary inputs To BCD
CALL	Call Subroutine
*	Comment
CDT	Correlated Data Table
ELSE	If/Then Function
ENDIF	If/Then Function
EXIT	Exit On Error
FTSR-IN	Fall Through Shift Register – In
FTSR-OUT	Fall Through Shift Register – Out
GOTO/LABEL	Go To/Label Function
IF/THEN	If/Then Functions
IMATH	Integer Math Operations
LEAD/LAG	Lead/Lag Operation
MATH	Real/Integer Math Operations
PACK	Pack Data
PACKAA	Pack Analog Alarm Data
PACKLOOP	Pack Loop Data
PRINT	Print Functions
RETURN	Return from SFPGM/SFSUB
SCALE	Scaling Values
SDT	Sequential Data Table
SSR	Synchronous Shift Register
UNSCALE	Unscaling Values

gram, whcih uses PACK, IMATH, MATH IF, ELSE and ENDIF statements for gain schedulin PID loop operation with gain scheduling.

A.1.4 Workstation-PLC serial communication

Data acquisition and control tasks can be accomplished to a relatively satisfactory level using the TI 545 PLC, 386/SX ATM, and other I/O modules installed in 505 base. Loops with variety of SF programs can be built using TISOFT running on 386/SX ATM. But there exist physical limitation due to the lack of the speed and expandability of the 386/SX ATM.

The 386/SX ATM provided with the 545 PLC is powered by 80386/SX chip with 16MHz clock speed, 4MB soldered(non-expandable) DRAM and 120MB hard disk drive(upgraded from original 40MB hard disk). Windows NT operating system requires at least 12MB main memory(DRAM) with 90MB hard disk memory. Borland C/C++ compiler requires 70MB hard disk space. The data acquisition and control tasks for rapid thermal CVD processor wafer temperature control needs a fast sampling rate(order of 0.1 second) and a powerful computing capability.

At the beginning of this research, additional unix workstaion equipped with 80386/33MHz CPU and SCO Unix operating system was provided. But the graphical monitoring software(TISTAR) was not flexible enough: the sampling rate was 1 second at the fastest, and the communication with 505 base was through TIWAY I(TI's LAN) with HDLC(High-level Data Link Control) communication protocol. Using the TISTAR on the workstaion with TI 545 PLC programming can only provide primitive graphical data display and a simple(PID) control environment.

So, additional external computing power with an operating system with a variety of software handling capability had been a necessary. A Pentium 120MHz, Windows NT operating system workstation(will be called as Newton W/S[2] hereafter) with 32MB DRAM, 1.0 GB hard disk, PCI bus ethernet card has been installed for that purpose. The Norton 6.0 Benchmark test showed CPU speed 9.0 for 386/SX and 410.0 for Pentium 120MHz(latter being about 45 times faster than former). Matlab for Windows®(version 4.0), Borland C/C^{++} for Windows(version

[2]named after 18th century Physicist, also the IP address of the workstation is newton.che.utexas.edu(128.83.162.132)

4.0) has been installed for data acquisition, identification and control programming.

The communication method chosen between the TI 545 PLC and Newton W/S was the RS-232C serial communication standard. PID loops and SF programs can be programmed on the 545 PLC using 386/ATM module and TISOFT. Then a supervisory program running on the Newton W/S sends variey of commands to the PID loops and writes necessary data on the PLC memories. The supervisory program also enables or disables the LOOP programs, and the sampling rate can be as fast as 10 Hz. Currently, 19,200 bps(bits per second) transfer rate is used for the serial communication.

In order for the Workstation and PLC to communicate data through serial interface and perform meaningful tasks inside the PLC, following items should be defined and provided:

- Serial communication method and a wire
- Data transmission protocol
- PLC task code

Serial Communication. In order for the two devices(a computer and a peripheral devices) to communicate each other, the two devices must be connected electronically. Two widely used communication methods are *parallel* and *serial* communication. In parallel communiction, a group of bits(8 bits = 1 byte = 1 character) move over several lines at the same time, and in serial communication, bits move one by one over a single line. If the two communicating devices are located closely(as in

a computer and a printer), parallel communcaiton can be used because it is fast and efficient. As the distance between the two devices increases, the multiple wires not only become more costly, but also the complexity of line drivers and receivers increases because of the difficulty of transmitting and receiving pulse signals over the long wires. Thus serial communication is used for data transmission over long distances.

In a simple serial communication, data can be transmitted back and forth through a single wire, but in real world, more wires are used to insure that the data transmission is being done in an orderly fashion. Most widely used serial interface is **RS-232** serial communication. The RS-232 is formally defined as *Recommended Standard Number 232*, from the Engineering Department of the Electronic Industries Association(EIA) in 1969[32]. Other proper name for RS-232 is "Interface Between Data Termianl Equipment[3] and Data Communication Equipment[4] Employing Serial Binary Data Interchange." Current most popular version of the RS-232 is revision C, which is formally referred to as RS-232C.

The RS-232C standard defines 25 communication lines, but only 11 of them are commonly implemented in real micorcomputer serial communciaton(see Table A.4). Also, IBM AT computers use only 9 lines subset of the original 25 lines(see Table A.5). The tables are extracted from [32].

The RS-232 standard also defines a number of electrical characteristics: total

[3]Computers are usually called Data Terminal Equipment(DTE)

[4]Peripheral devic that one is trying to communicate with, modems are called Data Communications Equipment(DCE)

Table A.4: A real world 25-pin RS-232 microcomputer serial interface

Pin	Abbreviation	name
1	—	Protective Ground
2	TD	Transmitted Data
3	RD	Received Data
4	RTS	Request to Send
5	CTS	Clear to Send
6	DSR	Data Set Ready
7	—	Signal Common
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready
22	RI	Ring Indicator
23	DSRD	Data Signal Rate Detector

cable capacitance should be less than 2500 picofrarads, a binary logic 1 is +5 to +15 volts for output lines and +3 to +15 volts for input lines, a binary logic 0 is -5 to -15 volts for output lines and -3 to -15 volts for input lines; voltages between -5 and +5 volts for for output lines and -3 and +3 volts for input lines are undefined.

Detail explanations on each communication lines can be found elsewhere[32][30][37], and are briefly shown below:

- **Clear to Send (CTS):** The CTS line is asserted(logic state 1) by the DCE when it is ready to receive data.
- **Data Carrier Detect (DCD):** The DCD line is asserted(logic state 1) whenever there is a data link in progress.
- **Data Set Ready (DSR):** The DSR line is asserted(logic state 1) by the DCE when it is ready to communicate wuith the DTE.
- **Data Terminal Ready (DTR):** The DTR line is asserted(logic state 1) by the DTE when it is ready to communicate iwth the DCE.

- **Received Data (RD):** The RD line is used by the DCE to send data to the DTE.
- **Ring Indicator (RI):** The RI line is asserted(logic state 1) by the DCE when a ring is detected.
- **Request to Send (RTS):** The RTS line is asserted(logic state 1) by the DTE when it wants to transmit data to the DCE.
- **Transmit Data (TD):** The TD line is used by the DTE to send data to the DCE.

In this research, to connect Newton W/S and the 545 PLC, modems were not needed. i.e., the length of the RS-232C cable is about 30 ft, and modems which are needed to connect the DTEs, which are located far apart(as far as the distance for which we have to use a phone to communicate each otehr) are not necessary. In this case, two DTEs(Newton W/S and the 545 PLC) are connected by a *null modem*, which fools the two DTEs think that they are connected by two modems.

545 PLC provides 9 pin IBM AT RS-232C serial interface, while Newton W/S has 25 pin RS-232C serial interface. The RS-232C 9 pin to 25 pin cable is shown in Figure **??**, which is introduced in [82]. As shown in the Figure **??**, only 3 wires need to be connected with a shield.

UART. Almost all micro computers use a special hardware device called a Universal Asynchronous Receiver Transmitter(UART) to implement an RS-232 serial

Table A.5: The IBM AT 9-pin RS232 serial interface

Pin	Abbreviation	name
1	DCD	Data Carrier Detect
2	RD	Received Data
3	TD	Transmit Data
4	DTR	Data Terminal Ready
5	—	Signal Common
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

interface. The IBM PC and compatibles use UARTs that are based on National Semiconductor's INS8250 family of UARTs. But most computers built today use the 16450 and 16550 UARTs. The 16450 UART is a higher speed version of the 8250 and the 16550 UART is a special version of the 16450 with FIFO(First In First Out) buffers.

Basically, a programmer has to know the function of the each register of the UART to do a serial communication programming. In this research, a system level Pascal code, which acceses the UART register and PC's input/output ports, was built and tested at the beginning(see Appendix C). But Windows operating system provides all the API(Application Program Interface)s necessary for serial communication programming. So, the system level programming handling the UART registers and input/ouput ports is not necessary at this point.

Example serial communication. Let's explain how a character data is converted into a group of bits and sent through a serial interface. The operating parameters used are 19200 bps transfer rate, odd parity, 1 stop bit, and uses serial port COM2.

Table A.6: Hexadeciaml digits and their correspondiong decimal values

Hexadecimal digit:	0	1	$\cdots$	9	A	B	C	D	E	F
Decimal value:	0	1	$\cdots$	9	10	11	12	13	14	15

In data acquisition and control programming, a *hexadecimal* number representation is frequently used(see Table A.6). On machines such as the 545 PLC, the task code is expressed as 2 hexadecimal digits[5].

For example,

$$\text{43h(h stands for hexadecimal) corresponds to } \underbrace{0100}_{4h} \quad \underbrace{0011}_{3h}$$

ANSI standard ASCII 7-bit code can represent 128 characters. Hexadecimal 43H(binary 0100 0011, 8 bits) or binary 1000011(7 bits) stands for ASCII code *C*. This single character is electrically converted to a digital signal(Figure A.9).

PLC Data Transmission Protocol. Through the RS-232 standard serial interface and using the null modem, one can communicate a character data to and from a DTEs connected each other. While RS-232 serial interface is a set of rules embodied in a piece of hardware(UART for example), a *protocol* is a set of rules defining the interactions between two machines. The TI 545 PLC uses either *Non-intelligent Terminal Protocol*[101] or *Transparent Byte Protocol*[102]. The former protocol is used in this work.

[5]545 PLC uses 8-bit bytes for internal programming; 1 *nibble* consists of 4 bits; thus 1 nibble represents 1 hexadecimal digit. For the communication between external device and the PLC, 7-bit ASCII code is used.

A subset of the standard 7-bit ASCII character set($2^7 = 128$ characters) is used in this protocol. The only characters which may form part of a valid message are those which can be used to form hexadecimal numbers(hexadecimal digits), i.e., the digits 0-9 and the letters A-F(as shown in Table A.6), plus the two characters which delimit the beginning and end of message. Following items constitute the valid message under the Non-intelligent Terminal Protocol:

- **Message delimiters:** A colon (:) delimits the beginning of the message, and a semicolon (;) delimits the end of the message. So, any characters between a colon and the next semicolon will be taken as a valid message, while any characters between a semicolon and a colon is ignored. Any characters other than the hexadecimal digits is considered as an error. When transmitting data to the external host(Newton W/S in this case), the PLC sends carriage return(CR) and line feed(LF) characters after the terminating semicolon.

- **Character count:** To aid in error control, the beginning-of-message colon is followed by a two-digit character count in hexadecimal notation. The character count is the total number of printable characters in the message, including the colon(1), character count(2), message body(mm), error-check code(4), and terminating semicolon(1). Thus character count(LL) is 8+mm and the default maximum of the LL upon the system initialization is 78 decimal(48h).

- **Body of the message:** This is a combination of the PLC task code, PLC memory addresses and data all hexadecimal representation. And the length of the message body is limited to LL−8.

- **Error-check code(ECC):** A four-character error-check code(ECC) forming a 16-bit hexadecimal number is included at the end of the message just before the semicolon delimiter. The ECC is a checksum, computed as follows:

  (1) Divide the cahracter count and the message body into four-character blocks, left-justfied and zero-filled.

  (2) Interprete each block as a four-digit hexadecimal number.

  (3) Add the resulting numbers.

  (4) The ECC is then the tow's complement of this sum.

For example, let's check the following unfinished message which needs ECC and the end of message delimiter(;),

:2E036009600B600D600A705D705E705F00460048

As explained previously, colon(:) informs start of message, '2E'(2Eh=46 decimal) is the length of the message, which in this case is 46 decimal including ':(1)',';(1)', '2E(2)' , task code '03(2)', and 9 of 4 byte word addresses(6009, 600B, 600D, 600A, 705D, 705E, 705F, 0046 and 0048) which sums up to 36 characters. So counting another 4 characters for the ECC, gives the total of 46 characters, which is 2Eh. ECC can be calculated as,

$$\begin{aligned} \text{ECC} &= \text{FFFFh - (2E03h + 6009h + 600Bh + 600Dh + 600Ah + 705Dh} \\ &\quad \text{+ 705Eh + 705Fh + 0046 h + 0048h) +0001h} \\ &= \text{002Ah.} \end{aligned}$$

So, the complete message constructed is

:2E036009600B600D600A705D705E705F00460048002A; (A.7)

```
    TISOFT
 DATE: 07-26-96     LOOP1  for Temp. Control, UT Chem. Eng.    545  LOOPALL   RN

     PID LOOP   1  TITLE: TCONTL-1              REMOTE SETPOINT: NONE
                                          CLAMP SP LIMITS:  LOW = +0.00000
   POS/VEL PID ALGORITHM: POS                               HIGH = +950.000
      LOOP VFLAG ADDRESS: V121
      SAMPLE RATE (SECS): +0.50000                    LOOP GAIN: +7.00000
                                           RESET (INTEGRAL TIME): +2.00000
PROCESS VARIABLE ADDRESS: NONE             RATE (DERIVATIVE TIME): +0.00000
         PV RANGE:  LOW = -220.000                  FREEZE BIAS: NO
                    HIGH = +1320.00
                                       DERIVATIVE GAIN LIMITING: NO
           PV IS BIPOLAR: NO              LIMITING COEFFICIENT: +10.0000
       SQUARE ROOT OF PV: NO
        20% OFFSET ON PV: NO             SPECIAL CALCULATION ON: PV
                                               SPECIAL FUNCTION: SFPGM1
     LOOP OUTPUT ADDRESS: NONE
       OUTPUT IS BIPOLAR: NO                      LOCK SETPOINT: NO
    20% OFFSET ON OUTPUT: NO                   LOCK AUTO/MANUAL: NO
                                                   LOCK CASCADE: NO
    RAMP/SOAK PROGRAMMED: NO
        RAMP/SOAK FOR SP: NO                     ERROR OPERATION: NONE
                                                  REVERSE ACTING: NO
          ALARM DEADBAND: +0.00000              MONITOR DEVIATION: NO
                                       DEVIATION ALARM: YELLOW = +100.004
   MONITOR LOW-LOW/HI-HI: NO                              ORANGE = +100.004
        MONITOR LOW/HIGH: NO
   PV ALARMS:   LOW-LOW = -0.02065       MONITOR RATE OF CHANGE: NO
                    LOW = -0.02065         RATE OF CHANGE ALARM: +200.000
                   HIGH = +950.015
               HIGH-HIGH = +974.992       MONITOR BROKEN XMITTER: NO


S-MEMORY AVAILABLE:29234                                              ENABLED
                                                                  545  LOOPALL   RN
 EXIT-F1   EDIT-F2                                             COMMNT-F7 EN/DIS-F8
```

Figure A.7: Loop programming table for loop 1

```
TITLE: LOOP1SF                                                SF PROGRAM    1
                    CONTINUE ON ERROR (Y,N): YES
               ERROR STATUS ADDR (Y,C,WY,V): C201
                     PROGRAM TYPE (N,P,C,R): RESTRICTED
                           CYCLE TIME (SEC): 0.0

00001 IF      T7 = 1
00002 PACK     TO/FROM TABLE.: FROM             TABLE ADDRESS.: V121
               NO. OF POINTS.: 3                DATA STRT ADDR: C11
00003 ENDIF
00004 IF      C11 = 1
00005 IMATH   LMN1 := V123
00006 IMATH   LMX1 := V125
00007 MATH    LTS1. := V19.
00008 IMATH   WY93 := LMN1
00009 ENDIF
00010 IF      C12 = 1
00011 MATH    LSP1. := V11.
00012 MATH    LKC1. := V13.
00013 MATH    LTI1. := V15.
00014 MATH    LTD1. := V17.
00015 MATH    LTS1. := V19.
00016 IMATH   WY93 := LMN1
00017 ENDIF
00018 MATH    LPV1. := WX9 * 0.48125 / 10 - 220
00019 MATH    V127 := WX9
00020 IF      LPV1. > 575.0
00021 MATH    LKC1. := V223.
00022 MATH    LTI1. := V243.
00023 ELSE
00024 IF      LPV1. > 525.0
00025 MATH    LKC1. := V221.
00026 MATH    LTI1. := V241.
00027 ELSE
00028 IF      LPV1. > 450.0
00029 MATH    LKC1. := V219.
00030 MATH    LTI1. := V239.
00031 ELSE
00032 IF      LPV1. > 400.0
00033 MATH    LKC1. := V217.
00034 MATH    LTI1. := V237.
00035 ELSE
00036 IF      LPV1. > 300.0
00037 MATH    LKC1. := V215.
00038 MATH    LTI1. := V235.
00039 ELSE
00040 IF      LPV1. > 200.0
00041 MATH    LKC1. := V213.
00042 MATH    LTI1. := V233.
00043 ELSE
00044 MATH    LKC1. := V211.
00045 MATH    LTI1. := V231.
00046 ENDIF
00047 ENDIF
00048 ENDIF
00049 ENDIF
00050 ENDIF
00051 ENDIF
00052 IF      LMN1 <= 0
00053 IMATH   LMN1 := 0
00054 IMATH   LMX1 := 0
00055 ENDIF
00056 MATH    V123 := LMN1
00057 MATH    V125 := LMX1
****  END ****

S-MEMORY AVAILABLE:29234                                          ENABLED
                                                          545  LOOPALL   RN
 EXIT-F1   EDIT-F2            FIND-F4  DELST-F5  INSST-F6 COMMNT-F7 EN/DIS-F8
```

Figure A.8: Program list for the SFPGM1

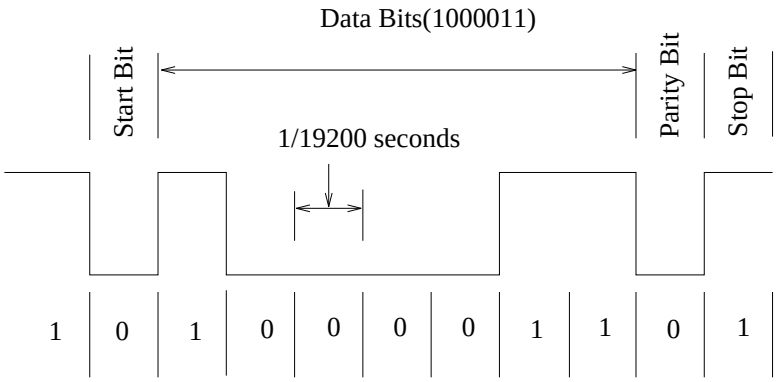


Figure A.9: Data bits for character *C*, with 19200bps, 7bits, odd parity and 1 stop parity communication parameter

Appendix B

Physical Constants and Derivations of Equations

B.1 Physical Constants

Table B.1: Some physical constants used

constant	description	value	unit
σ	Stefan-Boltzman constant	5.67051×10^{-8}	W/m^2K^{-4}
C_1	First radiation constant	3.7418×10^{10}	$\mu W \mu m^4 cm^{-2}$
C_2	Second radiation constant	14388	μmK
ρ	Density (S_i)	2329	K_g/m^3
Z	Thickness (S_i wafer)	725	μm
$C_p^{(a)}$	Specific heat of S_i/ T(K)		$J/K_g/K$
	280	691	
	300	713	
	400	785	
	500	832	
	> 500	$748 + 0.168T \pm 4$	
$k^{(b)}$	Thermal conductivity of S_i/ T (K)		$W/cm/K$
	300	1.56	
	400	1.05	
	500	0.80	
	600	0.64	
	700	0.52	
	800	0.43	
	900	0.36	
	1000	0.31	
	1100	0.28	
	1200	0.26	
	1300	0.25	

(a), (b) from 'Properties of Silicon' EMIS datareviews series. no. 4 London, INSPEC, 1988

B.2 Calculation of $\frac{\partial K_{ij}}{\partial T_i}$

Sensitivity of K_{ij} to temperature T can be obtained as follows. It will be derived using 2 steps. First, equation 3.36 is differentiated with respect to T with $\frac{\partial A_\epsilon}{\partial T}$ in a closed form. And as a second step, $\frac{\partial A_\epsilon}{\partial T}$ is derived and can be used to estimate $\frac{\partial K_{ij}}{\partial T_i}$.

From equation 3.36,

$$\frac{\partial K_{ij}}{\partial T_i} = \frac{\left(\frac{\partial F_{ij}}{\partial T_i}\right)(4\sigma D_{i,sum} T_i^3 + h_{w,i}) - F_{ij}(12\sigma D_{i,sum} {T_i}^2)}{(4\sigma D_{i,sum} T_i^3 + h_{w,i})^2} \tag{B.1}$$

From equation 3.24

$$\frac{\partial F}{\partial T} = \frac{\partial}{\partial T}(I - A_{\rho,\tau} V)^{-1} A_\epsilon W \tag{B.2}$$

For a given matrix X

$$\frac{dX^{-1}}{dT} = -X^{-1}\frac{dX}{dT}X^{-1} \tag{B.3}$$

Thus equation B.2 becomes

$$\frac{\partial F}{\partial T} = \frac{\partial}{\partial T}(I - A_{\rho,\tau} V)^{-1} A_\epsilon W + (I - A_{\rho,\tau} V)^{-1}\frac{\partial A_\epsilon}{\partial T} W \tag{B.4}$$

Using the fact that

$$\frac{\partial A_{\rho,\tau}}{\partial T} = \frac{\partial}{\partial T}(I - A_\epsilon) = -\frac{\partial A_\epsilon}{\partial T}$$

and using equation **??**eq:matinvderiv, equation B.4 can be simplified as

$$\frac{\partial F}{\partial T} = -(I - A_{\rho,\tau}V)^{-1}\frac{\partial A_\epsilon}{\partial T}\big(A_\epsilon V(I - A_{\rho,\tau}V)^{-1} + I\big)W \tag{B.5}$$

When the reflectivity of the silicon interfacing with air is R and absorption coefficient of silicon is α. Sato [72] has modeled the emissivity ϵ as

$$T_r = \exp(-\alpha d) \tag{B.6}$$

$$\epsilon = \frac{(1-R)(1-T_r)}{(1-RT_r)} \tag{B.7}$$

where T_r is the physical internal transmittance of the wafer and $d(cm)$ is the thickness of the wafer. Vandenabeele and Maex [103] tried to model the absorption coefficient α as

$$\alpha = s\lambda^p T^q \exp(-7000/T) \tag{B.8}$$

where λ is wavelength and p, q and s are fitting parameters. For wafers with rough back side surface, Vandenabeele and Maex [103] used modified T_r^* as

$$T_r^* = r\exp(-\alpha d) \tag{B.9}$$

So, emissivity ϵ can be estimated using equations B.7, B.8 and B.9. Derivative of emissivity with respect to temperature $\frac{\partial A_\epsilon}{\partial T}$ is

$$\frac{\partial A_\epsilon}{\partial T} = \frac{(1-R)dr\exp(-\alpha d)\Big(1-R+rR\exp(-\alpha d)\Big)\Big(\frac{\partial \alpha}{\partial T}\Big)}{(1-rR\exp(-\alpha d))^2} \tag{B.10}$$

where

$$\frac{\partial \alpha}{\partial T} = s\lambda^p T^{q-1}\exp(-7000/T)(q+7000/T) \tag{B.11}$$

So, the derivative of process gain with respect to temperature can be estimated using following set of 4 equations.

$$\frac{\partial K_{ij}}{\partial T_i} = \frac{\big(\frac{\partial F_{ij}}{\partial T_i}\big)(4\sigma D_{i,sum}T_i^3 + h_{w,i}) - F_{ij}(12\sigma D_{i,sum}{T_i}^2)}{(4\sigma D_{i,sum}T_i^3 + h_{w,i})^2} \tag{B.12}$$

$$\frac{\partial F}{\partial T} = -(I - A_{\rho,\tau}V)^{-1}\frac{\partial A_\epsilon}{\partial T}\big(A_\epsilon V(I-A_{\rho,\tau}V)^{-1}+I\big)W \tag{B.13}$$

$$\frac{\partial A_\epsilon}{\partial T} = \frac{(1-R)dr\exp(-\alpha d)\Big(1-R+rR\exp(-\alpha d)\Big)\Big(\frac{\partial \alpha}{\partial T}\Big)}{(1-rR\exp(-\alpha d))^2} \tag{B.14}$$

$$\frac{\partial \alpha}{\partial T} = s\lambda^p T^{q-1}\exp(-7000/T)(q+7000/T) \tag{B.15}$$

B.3 Notations Used for Matrix Operations (Chapter 6)

Some unusual operator notaions are used in Chapter 6. The Kronecker product of n $m \times n$ matrix $A = [a_{ij}]$ and a matrix B is denoted by $A \otimes B$ and is defined as

$$A \otimes B \equiv \begin{bmatrix} a_{11}B & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mn} \end{bmatrix} \tag{B.16}$$

The Hadamard product (Schur product, element-by-element product) is denoted by $A \circ B$ for two same size matrices of $m \times n$ and is defined as

$$A \circ B \equiv \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1n}b_{1n} \\ \vdots & & \vdots \\ a_{m1}b_{m1} & \cdots & a_{mn}b_{mn} \end{bmatrix} \tag{B.17}$$

For diagonal matrices D_1 and D_2, we have

$$\begin{aligned} D_1(A \circ B)D_2 &= (D_1 A D_2) \circ B \\ &= A \circ (D_1 B D_2) \\ &= (D_1 A) \circ (B D_2) \end{aligned} \tag{B.18}$$

We use

$$\text{vec}(A) \equiv [a_{11}a_{21} \cdots a_{m1}a_{12}a_{22} \cdots a_{mn}]^T \tag{B.19}$$

and it has the well-known identity (Horn and Johnson [38]),

$$\begin{aligned} \text{vec}(ABC) &= (C^T \otimes A)\text{vec}(B) \\ \text{vec}(A \circ B) &= \text{diagvec}(A))\text{vec}(B) \\ &= \text{diagvec}(B))\text{vec}(A) \end{aligned} \tag{B.20}$$

Various values are used for magnitude of matrices. Singular values and structured singular values are denoted by $\sigma(\cdot)$ and $\mu(\cdot)$, respectively. The matrix norms are

$$||A||_P \equiv \begin{cases} [\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^P]^{1/P} & \text{for } 1 \geq P < \infty \\ \max_{i,j} |a_{ij}| & \text{for } P = \infty \end{cases} \tag{B.21}$$

Then we have

$$||A||_2 = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2} = ||\text{vec}(A)||_2||A||_\infty = \max_{ij}|a_{ij}| = \text{vec}(A)||_\infty$$

Induced matrix norms are denoted as $||A||_{iP}$:

$$||A||_{iP} \equiv \max_{||x||_P=1} ||Ax||_P = \begin{cases} \max_i \sum_{j=1}^{m} |a_{ij}| & P = 1 \\ \sigma_{\max}(A) & P = 2 \\ \max_i \sum_{j=1}^{n} |a_{ij}| & P = \infty \end{cases}$$

For a nonsingular square matrix A, the condition number is denoted by $k(A)$ and is defined as

$$k(A) \equiv ||A||_2||A^{-1}||_2 = \sigma_{\max}(A)/\sigma_{\min}(A) \tag{B.22}$$

and the right minimally scaled condition number for a diagonal matrix D_2 is

$$k_R(A) \equiv \min k(AD_2) \tag{B.23}$$

For an ill-conditioned process, the condition number of its gain matrix is large. For multi-input multi-output processes, the process gain depends on the input and output directions. As in Skogestad et al.[86] and Li and Lee[52], we use the term "gain directionality" to refer to the singular values and corresponding right and left singular vectors of the gain matrix.

B.4 Proof of Nonsingularity of One-Row Perturbation

We assume, without loss of generality, that the first row of a matrix is perturbed. Then we have

$$\begin{aligned} G_P &= \begin{bmatrix} g_{11}(1+\Delta_{11}) & g_{12}(1+\Delta_{12}) & \cdots & g_{1n}(1+\Delta_{1n}) \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix} \\ &= \left(I + \begin{bmatrix} g_{11}\Delta_{11} & g_{12}\Delta_{12} & \cdots & g_{1n}\Delta_{1n} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} G^{-1} \right) G \\ &= \begin{bmatrix} 1+\sum_{j=1}^{n} \Delta_{1j}\lambda_{1j} & x & \cdots & x \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} G \end{aligned} \tag{B.24}$$

Nonsingularity condition 6.32 follows.

Appendix C

Data Acquisition and Control Program List

C.1 Windows C Program List for Data Acquisition and Control

```
// mimoman1.c
// This program does closed loop identification with PID control
// by PLC
//
#define STRICT
#define MBIG 1000000000
#define MSEED 161803398
#define MZ 0
#define FAC (1.0/MBIG)
#define MAX(x,y)   (((x) <= (y)) ? (y) : (x))
#define MIN(x,y)   (((x) >= (y)) ? (y) : (x))
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <math.h>
//#include "engine.h"
#pragma argused
#define ID_TIMER    1
#define LINES       3

struct PID_Param {
float set_pt[4];
float Kc[4];
float Ti[4];
float Td[4];
} loop;

typedef union int_or_float {
unsigned int  n[2];
float            x;
} number;

long FAR PASCAL _export WndProc (HWND, UINT, WPARAM, LPARAM);
unsigned int Hex_to_Dec(char *);
void Real_to_Hexa (int, float, char *);
void Check_Sum (char *);
void fMan_Mode(int);
void fAuto_Mode(int);
void fSet_PI(int);
void fSet_PD(int);
```

```
void fRead_IO_Var(char *, char *);
void fWrite_Only(char *szWrite_Com);
void fIdent_Step(int);
void fInt_to_Hexa(int, int, int, int, char *);
void fInt_to_Hexa_One(int, int, char *szBuf);
void fRecursive_Ident(char *);
void fChange_Ts(float);
void fAdd_Random(int, double);
void fSet_All_Setpt(void);
void fProg_Mode(void);
void fRun_Mode(void);
void fLoop_Enable(int);
void fLoop_Disable(int);
void fLoop_Bias_Write(int, float);     // Loop number, bias value
float ran3(long *);
int  to_Number(char ch);

char cInBuf[2], szWrite_Loop[4][100], szFile[30], step_pm = ' ';
char    szCommand_Line[100], szBuf2_M[100], s_dummy[20]; szBuf3[27];

static   int     nComID = -1 ;

UINT    cblength_M;
RECT    rcClient;
LPSTR        lpCmdLine1;
HINSTANCE    hInstance1;
HFILE        hFile;
HFILE        hFile_new;
PSTR szProgName = "MIMO Closed Loop Idntification";       // application name

float   Real_Sec, Real_Time, Set_Temp[3], Prep_Temp;
float   Init_Set_Temp[3], Delta_Set_Temp;
float   Kc_o[10][3],Kc_i[3],Ti_o[10][3],Ti_i[3],Td_o[3];
float   f_settle_time, f_ident_time, f_prep_time, f_input_time;
float   f_delta_temp[3], f_init_temp;
float   f_ramp_time, f_sub_total_time, f_ramp_real, f_ramp_scale, f_end_time;
float   f_ident_offset, T_Sample, f_Noise_Level;
float   f_sub_sub, f_subtract, fNoise_Time;
float   f1y_low_temp, f1y_high_temp;
float   f_time_0, f_time_1, f_time_2, f_time_3, f_time_4, f_time_5;
float   scale_x, f1_y_scale, f2_y_scale, f3_y_scale;
float   Control_Out_Range, Full_Temp_Range;
float   f_loop_time, f_stabilize_time, f_stable_offset, f_comp_time;
float   y_ramp_scale, tau_ramp, l_kc[3], l_taui[3];
////////////////////////////////////////////////////////////////////
//////////////////////
int     iTimer=0, iTime_Interval=200, iCheck = 0;
int     iDivider_Ident , iDivider_1 , n_high;
int     iAuto_Loop, u_random_sum=0;
int     iIdent_No=0, iTwx[9], iIdent_Check = 1;
int     iLoop_M, iMM, iLen_M;
int     iValue, u_ss_0, u_ss_1, u_ss_2;
int     f1xLeft=2000, f1yTop=18693, f1xRight=17680, f1yBottom=13295;
int     f2xLeft=2000, f2yTop=11295, f2xRight=17680, f2yBottom=7795;
int     f3xLeft=2000, f3yTop=5795,  f3xRight=17680, f3yBottom=2295;
int     f1_y_setPoint[3], f1_oy_setPoint[3], f3_y_setPoint[3], f3_oy_setPoint[3];
```

```
int     f1_sTemp_Wafer[4], f1_osTemp_Wafer[4], f2_sLamp_Out[3], f2_osLamp_Out[3];
int     f3_sTemp_Wafer[4], f3_osTemp_Wafer[4];
int     xClient, yClient, xPoint, oxPoint, yPoint;
int     xLabel[14], y1Label[3], y2Label[3], y3Label[5], fyAxUp[3], fyAxDn[3];
int     fxAxLeft, fxAxRight, fyAxL_R[3][11], Ident_Chk=0;
long    iSeed=-12345L;

double  u_ss=100.0, y_ss_T[3];

//Engine             *ep;
//Matrix             *prandom,*thm, *zz_vec;

int PASCAL WinMain( HINSTANCE hInstance,    // which program are we?
  HINSTANCE hPrevInst,    // is there another
one?
  LPSTR lpCmdLine,        // command line
arguments
  int nCmdShow)           // window size (icon, etc)
 {
 HWND            hWnd;                   // window handle from CreateWindow
 MSG             msg;                    // message from GetMessage
 WNDCLASS        wndclass;               // window class structure
 if(!hPrevInst)
  {
  wndclass.style         = CS_HREDRAW | CS_VREDRAW;  // style
  wndclass.lpfnWndProc   = WndProc;       // WndProc address
  wndclass.cbClsExtra    = 0;             // no extra class data
  wndclass.cbWndExtra    = 0;             // no extra window data
  wndclass.hInstance     = hInstance;     // which program?

// stock arrow cursor
  wndclass.hCursor       = LoadCursor (NULL, IDC_ARROW);

// stock blank icon
  wndclass.hIcon         = LoadIcon(NULL, IDI_APPLICATION);
  wndclass.lpszMenuName  = szProgName;    // menu name

// white background
  wndclass.hbrBackground = GetStockObject(WHITE_BRUSH);
  wndclass.lpszClassName = szProgName;    // window class name

  RegisterClass(&wndclass); // register the class
  }

 lstrcpy (lpCmdLine1, lpCmdLine);
//  hInstance1 = hInstance;

 hWnd = CreateWindow( szProgName,           // window class name
 szProgName,           // caption

WS_OVERLAPPEDWINDOW,  // style
 630,                  // default x
position
 0,                    // default y
position
 650,        // default width
```

```
 1000,        // default height
 NULL,                  // parent's
handle
 NULL,                  // menu
handle
 hInstance,             // which
program?
 NULL);                 // no init
data

// Data initialization
  lstrcpy (szCommand_Line,
//   ":2A03007E00880092007A0084008E60516052");
//    ":2E036009600B600D600A705D705E705F60516052");
  ":2E036009600B600D600A705D705E705F00460048");

  Check_Sum(szCommand_Line);

  for (iMM=0;iMM<3;++iMM)
  loop.set_pt[iMM]= 0.0;
//
hFile_new = _lcreat("C:\\Research\\ExpData\\tunepara.dat",0);
//
if(hFile_new == HFILE_ERROR)
{
MessageBox(NULL,"Error: Cannot create tuning parameter file",
  szProgName, MB_OK |
MB_ICONINFORMATION);
return NULL;
}
iLen_M = sprintf(szBuf2_M, "--------------------\n");
cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);
iLen_M= sprintf(szBuf2_M,
 "\n prep_ = %8.1f\n
input_ = %8.1f\nramp_ = %8.1f\nsettle_ = %8.1f\nident_ = %8.1f\nstabilize_ = %8.1f\n",
 f_prep_time,
 f_input_time,
 f_ramp_time,
 f_settle_time,
 f_ident_time,
 f_stabilize_time);
//
cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);
iLen_M = sprintf(szBuf2_M, "--------------------\n");
cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);
for (iLoop_M=0; iLoop_M<4;++iLoop_M)
 {
 iLen_M = sprintf(szBuf2_M,
 "Params for settling:
Kc_o[%d] = %6.3f  Ti_o[%d] = %6.3f  Td[%d] = %6.3f\n",
iLoop_M, Kc_o[iLoop_M],
iLoop_M, Ti_o[iLoop_M],
iLoop_M, Td_o[iLoop_M]);
cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);
 }
 iLen_M = sprintf(szBuf2_M, "--------------------\n");
```

```
 cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);

for (iLoop_M=0; iLoop_M<4;++iLoop_M)
 {
 iLen_M = sprintf(szBuf2_M,
 "Params for ident   :
Kc_i[%d] = %6.3f  Ti_i[%d] = %6.3f \n",
iLoop_M, Kc_i[iLoop_M],
iLoop_M, Ti_i[iLoop_M]);
  cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);
 }
iLen_M = sprintf(szBuf2_M, "--------------------\n");
cblength_M = _lwrite(hFile_new, szBuf2_M, iLen_M);
_lclose(hFile_new);

 while (!SetTimer (hWnd, ID_TIMER, iTime_Interval, NULL))
  if (IDCANCEL == MessageBox (hWnd,

"Too many clocks or timers!",

szProgName,

MB_ICONEXCLAMATION | MB_RETRYCANCEL))
  return FALSE ;

ShowWindow(hWnd, nCmdShow);                // make window visible
UpdateWindow (hWnd);

while ( GetMessage(&msg, NULL ,0,0) )      // message from Windows
  {
  TranslateMessage(&msg);               // convert keystrokes
  DispatchMessage(&msg);                // call windows procedure
  }
return msg.wParam;                         // return to Windows
}  // end WinMain


////////////////////////////////////////////////////////////////////
// main window procedure -- receives messages                     //
////////////////////////////////////////////////////////////////////
long FAR PASCAL _export WndProc(HWND hWnd,      // our window's handle
  UINT msg,
// message number
  WPARAM
wParam,  // word parameter
  LPARAM
lParam)  // long parameter
  {
  PAINTSTRUCT    ps ;
  HDC            hDC ;
  DCB            dcb ;
  UINT           cblength;      // characters written
  HBRUSH         hBrush;

  static  HPEN   hPen[5], hBlackPen[5];
```

```
  int             iLen, iLen3, iLoop, iK, jK, i_step, j_step;
  int             VMem_1, VMem_2;
  int             iKK, iJJ;

  float           Temp_Wafer[5],Loop_Output[3];
  float           Pyro_Sig[2], f_Time;

  char            szBuf[5],szBuf2[100], szBuf2_0[100], szBuf2_1[100];
  char            szBuf2_2[100], szBuf2_3[100];
  char            szBuf4[100], szRead_Buf[100], szScratch[100];
  char       cInBuf[2];
  char            szBuf_L[100];

 switch(msg)                        // which message?
 {
 case WM_CREATE :

// Scaling factors for PID parameters for ramping and settling
  n_high = 2;
  l_kc[0]=1.0;  l_kc[1]=1.0;   l_kc[2]=1.0;
  l_taui[0]=1.0;l_taui[1]=1.0; l_taui[2]=1.0;
// Kc values;
  Kc_o[0][0] = 3.80; Kc_o[0][1] = 5.50; Kc_o[0][2] = 5.00;
  Kc_o[1][0] = 3.70; Kc_o[1][1] = 5.10; Kc_o[1][2] = 5.00;
  Kc_o[2][0] = 3.60; Kc_o[2][1] = 5.00; Kc_o[2][2] = 5.00;
  Kc_o[3][0] = 3.50; Kc_o[3][1] = 4.50; Kc_o[3][2] = 5.00;
  Kc_o[4][0] = 3.40; Kc_o[4][1] = 4.00; Kc_o[4][2] = 4.90;
  Kc_o[5][0] = 3.30; Kc_o[5][1] = 3.50; Kc_o[5][2] = 4.80;
  Kc_o[6][0] = 3.20; Kc_o[6][1] = 3.40; Kc_o[6][2] = 4.70;
// Taui values;
  Ti_o[0][0] = 0.20 ; Ti_o[0][1] = 0.20; Ti_o[0][2] = 0.20;
  Ti_o[1][0] = 0.18 ; Ti_o[1][1] = 0.15; Ti_o[1][2] = 0.15;
  Ti_o[2][0] = 0.16 ; Ti_o[2][1] = 0.10; Ti_o[2][2] = 0.10;
  Ti_o[3][0] = 0.14 ; Ti_o[3][1] = 0.09; Ti_o[3][2] = 0.05;
  Ti_o[4][0] = 0.12 ; Ti_o[4][1] = 0.09; Ti_o[4][2] = 0.05;
  Ti_o[5][0] = 0.10;  Ti_o[5][1] = 0.09; Ti_o[5][2] = 0.05;
  Ti_o[6][0] = 0.09;  Ti_o[6][1] = 0.09; Ti_o[6][2] = 0.05;
// Multiply the PID parameters by scaling factor
  for (iKK=0;iKK<=2;++iKK)
  for (iJJ=1; iJJ<=n_high;++iJJ)
 {
 Kc_o[iJJ][iKK] = l_kc[iKK]*Kc_o[iJJ][iKK];
 Ti_o[iJJ][iKK] = l_taui[iKK]*Ti_o[iJJ][iKK];
 }
// PI parameters for identification (Pcontrol only)
  Kc_i[0] = 4.25;  Kc_i[1] = 5.75;   Kc_i[2] = 7.00;
  Ti_i[0] = 0.05;  Ti_i[1] = 0.04;   Ti_i[2] = 0.04;

//
  tau_ramp          = 4.00;
  Prep_Temp         = 100.0;
  Init_Set_Temp[0]  = 300.0;
  Init_Set_Temp[1]  = 300.0;
  Init_Set_Temp[2]  = 300.0;
  Control_Out_Range = 100.0;
  Full_Temp_Range   = Init_Set_Temp[0]+50.0;
```

```
  Delta_Set_Temp    = 5.0;
  f_prep_time    = 20.0;
  f_input_time   = 40.0;
  f_ramp_time    = 20.0;
  f_settle_time  = 60.0;
  f_ident_time   = 200.0;
  f_stabilize_time = 100.0;
  f_loop_time = f_ident_time + f_stabilize_time;
  Ident_Chk = 0;

  hPen[0] = CreatePen (PS_SOLID, 10, 0L) ;           // Black for setpoint
  hPen[1] = CreatePen (PS_SOLID, 10, RGB(255,0,0)) ; // Red for T1
  hPen[2] = CreatePen (PS_DASH, 10, RGB(0,255,0)) ;  // Green for T3
  hPen[3] = CreatePen (PS_DOT, 10, RGB(0,0,255)) ;   // Blue for T5
  hPen[4] = CreatePen (PS_SOLID, 10, RGB(255,0,255));// Magenta for T2

  hBlackPen[1] = hPen[1];
  hBlackPen[2] = hPen[2];
  hBlackPen[3] = hPen[3];

//   hBrush[0] = CreateSolidBrush (RGB(0,0,255));

//   f_Noise_Level  = 0.00;    // Random noise level between 0.0 and 1.0
  f1y_high_temp = MAX(MAX(Init_Set_Temp[0], Init_Set_Temp[1]),

MAX(Init_Set_Temp[0], Init_Set_Temp[2]))+25.0;
  f1y_low_temp  = MIN(MIN(Init_Set_Temp[0], Init_Set_Temp[1]),

MIN(Init_Set_Temp[0], Init_Set_Temp[2]))-10.0;
  f_time_0       = f_prep_time + f_input_time;
  f_time_1       = f_time_0 + f_ramp_time + f_settle_time;
  f_end_time     = f_time_1 + 4*f_loop_time + 600;
  f_time_2       = f_time_1 + f_loop_time;
  f_time_3       = f_time_2 + f_loop_time;
  f_time_4       = f_time_3 + f_loop_time;
          f_time_5       = f_time_4 + f_loop_time;
  f_ramp_real    = f_time_0 + f_ramp_time;
//   MessageBeep(0);
  oxPoint = f1xLeft;

  f1_y_scale = (f1yTop-f1yBottom)/(f1y_high_temp-f1y_low_temp);
  f2_y_scale = (f2yTop-f2yBottom)/Control_Out_Range;
  f3_y_scale = (f3yTop-f3yBottom)/(Init_Set_Temp[0] + 50.0);
  scale_x  = 1./f_end_time * ( f1xRight -f1xLeft );
//   Calculations of the x-axis labels
  xLabel[0] = f1xLeft + 0.0;
  xLabel[1] = f1xLeft +f_prep_time*scale_x;
  xLabel[2] = f1xLeft + f_time_0*scale_x;
  xLabel[3] = f1xLeft + (f_time_0+f_ramp_time)*scale_x;
  for (iKK=4; iKK<=12; ++iKK)
xLabel[iKK] = f1xLeft + (f_time_1+f_ident_time*(iKK-
4))*scale_x;
  xLabel[13] = f1xLeft + f_end_time*scale_x;
//
  fyAxUp[0] = f1yBottom + 100;
  fyAxDn[0] = f1yBottom - 100;
```

```
  fyAxUp[1] = f2yBottom + 100;
  fyAxDn[1] = f2yBottom - 100;
  fyAxUp[2] = f3yBottom + 100;
  fyAxDn[2] = f3yBottom - 100;
// Y axix tics position calculation
  fxAxLeft = 1900; fxAxRight = 2100;
  for (iKK=0; iKK<11; ++iKK)
  fyAxL_R[0][iKK] = f1yBottom + (f1yTop-f1yBottom)/10*iKK;
  for (iKK=0; iKK<=10; ++iKK)
  fyAxL_R[1][iKK] = f2yBottom + (f2yTop-f2yBottom)/10*iKK;
  for (iKK=0; iKK<=10; ++iKK)
  fyAxL_R[2][iKK] = f3yBottom + (f3yTop-f3yBottom)/10*iKK;
//
  for (iKK=0; iKK<4; ++iKK)
  {
  f1_osTemp_Wafer[iKK] = f1yBottom;
  f3_osTemp_Wafer[iKK] = f3yBottom;
  Temp_Wafer[iKK] = 0.0;
  }
  for (iKK=0; iKK<3; ++iKK)
  {
  f1_oy_setPoint[iKK] = f1yBottom;
  f3_oy_setPoint[iKK] = f3yBottom;
  f2_osLamp_Out[iKK] = f2yBottom;
  loop.set_pt[iKK] = 0.0;
  }
  Real_Sec = iTime_Interval /1000.;
//   MessageBeep(0);
  iDivider_1 = ceil(1.0/ Real_Sec);
  iDivider_Ident = 1;
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Open communication port
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
  nComID = OpenComm ("COM2", 128, 128) ;
  if (nComID < 0)
  MessageBox (hWnd,
  "Could not open COM2",
  "Warning",
  MB_OK) ;
  else
  {
  FlushComm (nComID, 0) ; /* empty output queue */
  FlushComm (nComID, 1) ; /* empty input queue  */
  BuildCommDCB ("COM2:19200,o,7,1", &dcb) ;
  SetCommState (&dcb);
  EscapeCommFunction (nComID, RESETDEV) ;
  MessageBox (hWnd,
  "COM2 is open.",
  "Message",
  MB_OK);
//  Put the PI Loop into a manual mode & create a file
  fMan_Mode(0); fMan_Mode(1); fMan_Mode(2);
  wsprintf(szFile, "C:\\Research\\ExpData\\O%d00.DAT\0",iIdent_No + 1);
  hFile = _lcreat(szFile,0);
```

```
//
  MessageBox (NULL,
 "PID controllers are in manual mode, you can turn on the lamp s/w",
 "OK Message",
 MB_OK);
// Enable and disable loops
//   fLoop_Disable(1); fLoop_Disable(2); fLoop_Disable(3);
//   fLoop_Enable(1);  fLoop_Enable(2);  fLoop_Enable(3)
//
// Write Kc and Taui values to V memories
//
  for (iKK=0; iKK<=n_high; ++iKK)
 for (iJJ=0; iJJ<=2; ++iJJ)
  {
  lstrcpy(szBuf_L, ":1A02");
  Real_to_Hexa(211+iKK*2+40*iJJ, Kc_o[iKK][iJJ], szBuf2);
  lstrcat(szBuf_L,szBuf2);
  Check_Sum(szBuf_L);
  fWrite_Only(szBuf_L);
  lstrcpy(szBuf_L, ":1A02");
  Real_to_Hexa(231+iKK*2+40*iJJ, Ti_o[iKK][iJJ], szBuf2);
  lstrcat(szBuf_L,szBuf2);
  Check_Sum(szBuf_L);
  fWrite_Only(szBuf_L);
  }
 }
 break ;
 case WM_TIMER :
  Real_Time = Real_Sec * iTimer;
  if ((Real_Time == f_end_time) || (Real_Time< 1.0 ) ||
  ((Real_Time > f_time_5) && (Real_Time < f_end_time)))
  {
  for (iKK=0; iKK<3; ++iKK)
  {
  fMan_Mode(iKK);
  fInt_to_Hexa_One(123+iKK*10,0, szBuf2);
  fWrite_Only(szBuf2);
  fInt_to_Hexa_One(125+iKK*10,0, szBuf2);
  fWrite_Only(szBuf2);
  }
  }
  else if (Real_Time > f_end_time)
{
MessageBeep(0);MessageBeep(0);
KillTimer(hWnd, ID_TIMER);
}
f_ident_offset = Real_Time - f_time_1;
if( f_ident_offset <= 0.0 )
 f_ident_offset = 0.0;
j_step = floor(f_ident_offset/f_loop_time);
f_comp_time = j_step * f_loop_time;
f_Time = f_time_1 + f_comp_time;
f_stable_offset = Real_Time - f_Time;

// Prepare data when time = f_prep_time
if (Real_Time == f_prep_time)
```

```
{
  MessageBeep(0);
  fChange_Ts(0.2);
  for (iKK=0; iKK<3; ++iKK)
  Set_Temp[iKK]=Prep_Temp;
  for (iKK=0; iKK<3; ++iKK)
  {
  loop.Kc[iKK]=Kc_o[0][iKK];
loop.Ti[iKK]=Ti_o[0][iKK];
  loop.set_pt[iKK]=Set_Temp[iKK];
  fSet_PI(iKK);
  fAuto_Mode(iKK);
  }
 }
// Prepare data for set point ramping up
 else if (Real_Time == f_time_0)
 {
  MessageBeep(0);
  f_init_temp = Prep_Temp;
  for (iKK=0; iKK<=2; ++iKK)
 f_delta_temp[iKK] = Init_Set_Temp[iKK] -
f_init_temp;
 }
f_subtract = Real_Time - f_time_0;

if ((Real_Time >= f_time_0) && (Real_Time < f_ramp_real)
  && ((iTimer % iDivider_1) == 0))
{
f_ramp_scale = f_subtract/f_ramp_time;
// f_ramp_scale = f_subtract/tau_ramp;
// y_ramp_scale = 1-(1+f_ramp_scale)*exp(-f_ramp_scale);
for (iKK=0;iKK<=2;++iKK)

Set_Temp[iKK]=f_ramp_scale*f_delta_temp[iKK]+f_init_temp;
fSet_All_Setpt();
}
else if (f_subtract == f_ramp_time)
 {
 MessageBeep(0);
 for (iKK=0;iKK<=2;++iKK)
  Set_Temp[iKK] = Init_Set_Temp[iKK];
 fSet_All_Setpt();
 }
else if ((Real_Time == f_Time) && (Real_Time < f_time_5))
{
 MessageBeep(0);
 for (iKK=0; iKK<=2; ++iKK)
 {
  lstrcpy(szBuf_L, ":1A02");
  Real_to_Hexa(211+n_high*2+iKK*40,
Kc_i[iKK], szBuf2_M);
  lstrcat(szBuf_L,szBuf2_M);
  Check_Sum(szBuf_L);
  fWrite_Only(szBuf_L);
 //
  lstrcpy(szBuf_L, ":1A02");
```

```
  Real_to_Hexa(231+n_high*2+iKK*40,
Ti_i[iKK], szBuf2_M);
  lstrcat(szBuf_L,szBuf2_M);
  Check_Sum(szBuf_L);
  fWrite_Only(szBuf_L);
 }

 fIdent_Step(j_step);
 Ident_Chk = 1;
}
else if (Real_Time == f_time_5)
{
for (iKK=0; iKK<3; ++iKK)
{
loop.set_pt[iKK] = 20.0;
fSet_PI(iKK);
}
}

else if ((f_stable_offset == f_ident_time) && (Real_Time <
f_time_5))
{
  Ident_Chk = 1;
  MessageBeep(0);
  for (iKK=0; iKK<3; ++iKK)
  {
//
  loop.set_pt[iKK] = Init_Set_Temp[iKK];
  fSet_PI(iKK);
//
// Write VKc and VTi(zero) values to the V memory
//
  lstrcpy(szBuf_L, ":1A02");
  Real_to_Hexa(211+n_high*2+iKK*40,
Kc_o[n_high][iKK], szBuf2_M);
  lstrcat(szBuf_L,szBuf2_M);
  Check_Sum(szBuf_L);
  fWrite_Only(szBuf_L);
//
                      lstrcpy(szBuf_L, ":1A02");
  Real_to_Hexa(231+n_high*2+iKK*40,
Ti_o[n_high][iKK], szBuf2_M);
  lstrcat(szBuf_L,szBuf2_M);
  Check_Sum(szBuf_L);
  fWrite_Only(szBuf_L);
//   loop.Kc[iKK]     = Kc_o[iKK];
//   loop.Ti[iKK]     = Ti_o[iKK];

  fAuto_Mode(iKK);
  fChange_Ts(0.2);
}
  }
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Read the data from the process
////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////

fRead_IO_Var(szCommand_Line, szRead_Buf);
for (jK=0; jK < 9; jK += 1)
{
  for (iK=0; iK<4; iK += 1)
  szBuf[iK] = szRead_Buf[4*jK+iK+5];
  szBuf[iK] ='\0';
  iTwx[jK] = Hex_to_Dec(szBuf);
}
for (iKK=0; iKK<4; ++iKK)
 Temp_Wafer[iKK] = iTwx[iKK]*0.048125 -220.0; // Scaled
conversion

 // for the T/C input

// module(not Eng. conversion).
Loop_Output[0] = iTwx[4]/320.;
Loop_Output[1] = iTwx[5]/320.;
Loop_Output[2] = iTwx[6]/320.;
Pyro_Sig[0] = iTwx[7]/32000.;
Pyro_Sig[1] = iTwx[8]/32000.;

iLen = sprintf(szBuf2,
  "%5.1f %5.2f %5.2f %5.2f %6.2f %6.2f %6.2f %6.2f %d %d %d %d %d
\n",
Real_Time,
loop.set_pt[0], loop.set_pt[1], loop.set_pt[2],
Temp_Wafer[0],Temp_Wafer[1],Temp_Wafer[2], Temp_Wafer[3],
iTwx[4],iTwx[5],iTwx[6],iTwx[7],iTwx[8]);
sprintf(szBuf2_0,
"%5.1f sec.          Loop 1    Loop 2    Loop 3\n", Real_Time);
sprintf(szBuf2_1,
"  Set point   =>    %5.1f  %5.1f   %5.1f       Pyro. sig. =>   %6.4f  %6.4f \n",
loop.set_pt[0], loop.set_pt[1], loop.set_pt[2],
Pyro_Sig[0], Pyro_Sig[1]);

sprintf(szBuf2_2,
 "  TC            =>     %6.2f   %6.2f   %6.2f   %6.2f\n",
Temp_Wafer[0],Temp_Wafer[1],Temp_Wafer[2], Temp_Wafer[3]);

sprintf(szBuf2_3,
 "  Contr. out  =>    %6.2f  %6.2f  %6.2f \n",
Loop_Output[0],Loop_Output[1],Loop_Output[2]);

if(   (((Real_Time >= f_prep_time)&&(Real_Time < f_time_1)) ||
   ((Real_Time >= f_time_1)&&( Ident_Chk == 0))) &&
   ((iTimer % iDivider_1) == 0) )
{
 cblength = _lwrite(hFile, szBuf2,iLen);
 if(cblength == (UINT)HFILE_ERROR)
{
 MessageBox(NULL, "Error: Cannot write to file",
  szProgName, MB_OK |
MB_ICONINFORMATION);
  return NULL;
```

```
}
 }
// if ( (((Real_Time >= f_time_1) && (Ident_Chk == 1)) &&
//   ((iTimer % iDivider_Ident) == 0)) ||
//    (Real_Time > f_time_5 ))
// {
            else
{
     cblength = _lwrite(hFile, szBuf2,iLen);
 if(cblength == (UINT)HFILE_ERROR)
{
 MessageBox(NULL, "Error: Cannot write to file",
  szProgName, MB_OK |
MB_ICONINFORMATION);
  return NULL;
}
}


// Put the estimated parameter to a file
if ((iTimer % iDivider_1) == 0)
{
// MessageBeep(0);
hDC = GetDC(hWnd);
// Write process data on the top part of the screen
TextOut (hDC, 5, 15, szBuf2_0, lstrlen(szBuf2_0) ); // Real time
TextOut (hDC, 5, 30, szBuf2_1, lstrlen(szBuf2_1) ); // print the process data
TextOut (hDC, 5, 45, szBuf2_2, lstrlen(szBuf2_2) ); // print the process data
TextOut (hDC, 5, 60, szBuf2_3, lstrlen(szBuf2_3) ); // print the process data
// Draw the trend data and the setpoints

GetClientRect(hWnd, &rcClient);        // get client dimensions
yClient = rcClient.bottom;             // get height of window
SetMapMode(hDC, MM_HIMETRIC);
SetViewportOrg(hDC, 0, yClient);
xPoint = f1xLeft + Real_Time*scale_x;

// Figure 1 drawing: Temperature response in detail
  for (iKK=0; iKK<3; ++iKK)
  {
 f1_y_setPoint[iKK] = f1yBottom + (loop.set_pt[iKK]-f1y_low_temp)*f1_y_scale;
  if (loop.set_pt[iKK] > f1y_low_temp)
 {
 SelectObject (hDC, hBlackPen[iKK+1]);
 MoveTo(hDC, oxPoint, f1_oy_setPoint[iKK]) ;
 LineTo(hDC, xPoint, f1_y_setPoint[iKK]) ;
 }
  else
 {
 f1_y_setPoint[iKK] = f1yBottom;
 }
 if (Temp_Wafer[iKK] > f1y_low_temp)
 {
 if (Temp_Wafer[iKK] < f1y_high_temp)
f1_sTemp_Wafer[iKK]=f1yBottom+(Temp_Wafer[iKK]-
f1y_low_temp)*f1_y_scale;
```

```
 else if (Temp_Wafer[iKK] >= f1y_high_temp)
f1_sTemp_Wafer[iKK]=f1yTop;

 SelectObject (hDC, hPen[iKK+1]);
 MoveTo(hDC, oxPoint, f1_osTemp_Wafer[iKK] ) ;
 LineTo(hDC, xPoint, f1_sTemp_Wafer[iKK]);
 }
//  MessageBeep(0);
  else
 f1_sTemp_Wafer[iKK] = f1yBottom;
  }

// Figure 2 drawing: Controller output
  for (iKK=0; iKK<3;  ++iKK)
  {
  f2_sLamp_Out[iKK] = f2yBottom + Loop_Output[iKK]*f2_y_scale;
  if (f2_sLamp_Out[iKK] >= f2yTop)
      f2_sLamp_Out[iKK] = f2yTop;
  SelectObject (hDC, hPen[iKK+1]);
  MoveTo(hDC, oxPoint, f2_osLamp_Out[iKK]);
  LineTo(hDC, xPoint, f2_sLamp_Out[iKK]);
  }
// Figure 3 drawing: responses in larger scale
  for (iKK=0;  iKK<3; ++iKK)
  {
f3_y_setPoint[iKK] = f3yBottom +
loop.set_pt[iKK]*f3_y_scale;
SelectObject (hDC, hBlackPen[iKK+1]);
MoveTo(hDC, oxPoint, f3_oy_setPoint[iKK]) ;
LineTo(hDC, xPoint, f3_y_setPoint[iKK]) ;
f3_sTemp_Wafer[iKK] = f3yBottom +
Temp_Wafer[iKK]*f3_y_scale;
SelectObject (hDC, hPen[iKK+1]);
MoveTo(hDC, oxPoint, f3_osTemp_Wafer[iKK] ) ;
LineTo(hDC, xPoint, f3_sTemp_Wafer[iKK]);
}

ReleaseDC(hWnd,hDC);
// InvalidateRect (hWnd, NULL, FALSE);
}
oxPoint = xPoint;
for (iKK=0; iKK<3; ++iKK)
{
f1_oy_setPoint[iKK] = f1_y_setPoint[iKK];
f2_osLamp_Out[iKK] = f2_sLamp_Out[iKK];
f3_oy_setPoint[iKK] = f3_y_setPoint[iKK];
}
for (iKK=0; iKK<4; ++iKK)
{
f1_osTemp_Wafer[iKK] = f1_sTemp_Wafer[iKK];
f3_osTemp_Wafer[iKK] = f3_sTemp_Wafer[iKK];
}
iTimer +=1;


  return 0;
```

```
 case WM_PAINT:
  hDC = BeginPaint (hWnd, &ps) ;

  hBrush = GetStockObject (NULL_BRUSH) ; // get the handle to LTGRAY_Brush
  SelectObject (hDC, hBrush) ;            // Select the handle to device context
  GetClientRect(hWnd, &rcClient);        // get client dimensions
  yClient = rcClient.bottom;             // get height of window
  SetMapMode(hDC, MM_HIMETRIC);
  SetViewportOrg(hDC, 0, yClient);
  Rectangle(hDC, f1xLeft, f1yTop, f1xRight, f1yBottom);
  Rectangle(hDC, f2xLeft, f2yTop, f2xRight, f2yBottom);
  Rectangle(hDC, f3xLeft, f3yTop, f3xRight, f3yBottom);
// Figure 1 title
  hBrush = CreateSolidBrush (RGB(12,1,155));
  SelectObject (hDC, hBrush) ;
  Rectangle(hDC, f1xLeft+4000, f1yTop+1000, f1xRight-4000, f1yTop+250);
  DeleteObject (hBrush) ;
  sprintf(szBuf," Temperatures vs. time (detail) ");
  TextOut(hDC, f1xLeft+4800, f1yTop+865, szBuf, lstrlen(szBuf));
// Figure 2 title
//   hBrush = GetStockObject (GRAY_BRUSH);
  hBrush = CreateSolidBrush (RGB(0,136,0));
  SelectObject (hDC, hBrush) ;
  Rectangle(hDC, f1xLeft+4000, f2yTop+800, f1xRight-4000, f2yTop+50);
  DeleteObject (hBrush) ;
  sprintf(szBuf," Controller output vs. time ");
  TextOut(hDC, f1xLeft+5250, f2yTop+665, szBuf, lstrlen(szBuf));
// figure 3 title
  hBrush = CreateSolidBrush (RGB(142,35,107));
  SelectObject (hDC, hBrush) ;
  Rectangle(hDC, f1xLeft+4000, f3yTop+800, f1xRight-4000, f3yTop+50);
  DeleteObject (hBrush) ;
  sprintf(szBuf," Temperature vs. time (full) ");
  TextOut(hDC, f1xLeft+5250, f3yTop+665, szBuf, lstrlen(szBuf));
// Bottom box
  hBrush = CreateSolidBrush (RGB(255,128,0));
  SelectObject (hDC, hBrush) ;
  Rectangle(hDC, 500, 1500, f1xRight+500, 100);
  DeleteObject (hBrush) ;
  lstrcpy(szScratch,
  " University of Texas at Austin, Chemical Engineering,CPE 5.420-- Wonhui Cho ");
  TextOut(hDC, 1500, 1050, szScratch, lstrlen(szScratch));
// Drawing x-ticss and labels for x axis
  for (jK=0; jK<=2; ++jK)
  {
  SelectObject (hDC, hPen[0]);
  for (iKK=0; iKK<13; ++iKK)
  {
  MoveTo(hDC, xLabel[iKK], fyAxDn[jK]);
  LineTo(hDC, xLabel[iKK], fyAxUp[jK]);
      sprintf(szBuf, "%5.0f", f_time_0 + f_ramp_time);
      TextOut(hDC,xLabel[3]-500,fyAxDn[jK]-60,szBuf,lstrlen(szBuf));
  }
      for (iKK=0; iKK<9; ++iKK)
  {
  sprintf(szBuf,"%5.0f", f_time_1+f_ident_time*iKK);
```

```
  TextOut(hDC,xLabel[4+iKK]-600,fyAxDn[jK]-60,szBuf,lstrlen(szBuf));
  }
  sprintf(szScratch, "(sec)");
  TextOut(hDC,xLabel[13]-400,fyAxDn[jK]-550,szScratch,
lstrlen(szScratch));
}

// Drawing y-tics and labels for yaxis
  for  (jK=0; jK<=2; ++jK)
  {
  SelectObject (hDC, hPen[0]);
  for (iKK=0; iKK<11; ++iKK)
  {
  MoveTo(hDC, fxAxLeft, fyAxL_R[jK][iKK]);
  LineTo(hDC, fxAxRight,fyAxL_R[jK][iKK]);
  }
  }
// Labelling y-tics for Figure 1
  for  (iKK=0; iKK<3; ++iKK)
  {
sprintf(szScratch, "%5.1f", f1y_low_temp +
 iKK*(f1y_high_temp-
f1y_low_temp)/2.0);
TextOut(hDC, fxAxLeft-1200, fyAxL_R[0][iKK*5]+300,
szScratch,
 lstrlen(szScratch));
  }
  lstrcpy(szScratch, "(deg. C)");
  TextOut(hDC, 2000, f1yTop+665, szScratch, lstrlen(szScratch));

// Labelling y-tics for Figure 2
  for  (iKK=0; iKK<3; ++iKK)
  {
sprintf(szScratch, "%5.1f",iKK*Control_Out_Range/2.0);
TextOut(hDC, fxAxLeft-1200, fyAxL_R[1][iKK*5]+300,
szScratch,
 lstrlen(szScratch));
  }
  lstrcpy(szScratch, "(%,control signal)");
  TextOut(hDC, 2000, f2yTop+665, szScratch, lstrlen(szScratch));

// Labelling y-tics for Figure 3
  for  (iKK=0; iKK<3; ++iKK)
  {
sprintf(szScratch, "%5.1f",iKK*Full_Temp_Range/2.0);
TextOut(hDC, fxAxLeft-1200, fyAxL_R[2][iKK*5]+300,
szScratch,
 lstrlen(szScratch));
  }
  lstrcpy(szScratch, "(deg. C)");
  TextOut(hDC, 2000, f3yTop+665, szScratch, lstrlen(szScratch));
  EndPaint (hWnd, &ps) ;

  return 0;
 case WM_DESTROY :             // we handle this message
KillTimer (hWnd, ID_TIMER);
```

```
for (iLoop = 0; iLoop < 3; ++iLoop)
{
Set_Temp[iLoop] = 0.0;
fAuto_Mode(iLoop);
}
fSet_All_Setpt();
for (iLoop = 0; iLoop < 3; ++iLoop)
{
fMan_Mode(iLoop); // put the PID controller to manual
VMem_1 = 10*(iLoop+1) + 113;
VMem_2 = 10*(iLoop+1) + 115;
fInt_to_Hexa(VMem_1,0,VMem_2,0, szBuf2);
fWrite_Only(szBuf2);
// fAdd_Random (iLoop, 0.0);
}
 CloseComm (nComID);        // mode and make the output 0
//  engClose(ep);
//  mxFreeMatrix(thm);
//  mxFreeMatrix(zz_vec);
 for (iKK = 0; iKK<5; ++iKK)
 DeleteObject (hPen[iKK]);
 _lclose(hFile);
PostQuitMessage(0);        // send WM_QUIT
  return 0;
  }   // end switch(msg)
  return DefWindowProc (hWnd, msg, wParam, lParam) ;
  }  // end WndProc

//////////////////////////////////////////////////////////////////
//
//    to_Number()
//
//////////////////////////////////////////////////////////////////
int to_Number(char ch)
  {
  /* return decimal value for one hexadecimal */

  if (ch>='0' && ch<='9')
  return (ch-'0');

  if (ch>='A' && ch<='F')
  return (10+ch-'A');

  if (ch>='a' && ch<='f')
  return (10+ch-'a');

  else
  return (0);
  }
//////////////////////////////////////////////////////////////////
//
//    Hex_to_Dec()
//
//////////////////////////////////////////////////////////////////
unsigned int Hex_to_Dec(char *hexa)
  {
```

```
  /* 'hexa' is a 4-place hexadecimal */

  int i;
  long int sum=0;

  for (i=0; i<4; ++i)
  sum = sum*16 + to_Number(hexa[i]);
  return sum;
  }
////////////////////////////////////////////////////////////////
//
//   Function for Real to Hexadeciaml Conversion
//
////////////////////////////////////////////////////////////////
void Real_to_Hexa(int Vmem_No, float fValue, char *szBuf)
{
number temp;
temp.x = fValue;
wsprintf (szBuf, "%04X%04X%04X%04X\0",
Vmem_No-1, temp.n[1], Vmem_No, temp.n[0]);
}

////////////////////////////////////////////////////////////////
//
//   Function for checksum calculation
//
////////////////////////////////////////////////////////////////
void Check_Sum (char *szLine)
{
int iChar, iChk,iChlim, iFour;
unsigned long int iCheck_Sum;
char szCheck[5], szCheck_Sum[6];
iChk = (strlen(szLine) - 1)/4 -1;
iCheck_Sum = 4294967295;
//printf("The initial value of iCheck_Sum= %lu\n", iCheck_Sum);
for (iFour = 0; iFour <= iChk; ++iFour)
{
  iChlim=iFour*4+1;
  for (iChar =iChlim; iChar <iChlim+4; ++iChar)
szCheck[iChar-iChlim] = szLine[iChar];
  szCheck[iChar-iChlim] = '\0';
  iCheck_Sum += -Hex_to_Dec(szCheck);
  //   printf("szCheck = %s  dec(szCheck)= %06lu iCheck_Sum = %lu\n",
// szCheck, Hex_to_Dec(szCheck),iCheck_Sum);
}
wsprintf(szCheck_Sum,"%04X",((iCheck_Sum+1) % 65536));
//itoa(((iCheck_Sum + 1) % 65536), szCheck_Sum,16);
for (iChar=0;iChar<4;++iChar)
  szCheck_Sum[iChar]=toupper(szCheck_Sum[iChar]);
lstrcat(szLine,szCheck_Sum);
lstrcat(szLine,";\0");
}
////////////////////////////////////////////////////////
//
// Function to set the iLoop'th Loop in manual mode
```

```
//
//////////////////////////////////////////////////////
void fMan_Mode(int iLoop_No)
{
char szMan_Mode[20];

switch(iLoop_No)
  {
  case 0 :
  lstrcpy(szMan_Mode, ":120200788000\0");
  break ;
  case 1 :
  lstrcpy(szMan_Mode, ":120200828000\0");
  break ;
  case 2 :
  lstrcpy(szMan_Mode, ":1202008C8000\0");
  break ;
  case 3 :
  lstrcpy(szMan_Mode, ":120200968000\0");
  break ;

  }
  Check_Sum(szMan_Mode);
  fWrite_Only(szMan_Mode);

}
//////////////////////////////////////////////////////
//
// Function to set the iLoop'th Loop in Auto mode
//
//////////////////////////////////////////////////////
void fAuto_Mode(int iLoop_No)

{
char szAuto_Mode[20];
switch(iLoop_No)
  {
  case 0 :
  lstrcpy(szAuto_Mode, ":120200785000\0");
  break ;
  case 1 :
  lstrcpy(szAuto_Mode, ":120200825000\0");
  break ;
  case 2 :
  lstrcpy(szAuto_Mode, ":1202008C5000\0");
  break ;

  }
  Check_Sum(szAuto_Mode);
  fWrite_Only(szAuto_Mode);
}
//////////////////////////////////////////////////////////////
//
// Function to write the command to the PLC and just check the
// response from the PLC
//
```

```
////////////////////////////////////////////////////////////////
void fWrite_Only(char *szWrite_Com)
{
int iChar, nReadChars, nStatus;
char szRead_Sum[40];
 nStatus = WriteComm (nComID,
 szWrite_Com,
 lstrlen(szWrite_Com));

 if (nStatus < 0)
 MessageBox (NULL,
 "Output Comm error.(Write loop)",
 "Message",
 MB_OK);
 else
 lstrcpy(szRead_Sum, """");
 iChar = 0;
 do
 {
 rcomm0:
 nReadChars = ReadComm (nComID, cInBuf, 1);
 if (nReadChars != 1) goto rcomm0;
  szRead_Sum[iChar]=cInBuf[0];
  iChar += 1;
 }
 while (cInBuf[0]!=13) ;

}
////////////////////////////////////////////////////////////////
//
// Set PID parameters of iLoopth PID controller)
//
////////////////////////////////////////////////////////////////
void fSet_PI(int iLoop)
{
char szBuf_Loop[17];
  lstrcpy(szWrite_Loop[iLoop],":3A02\0");

// 74 characters with write mode
  Real_to_Hexa(10*(iLoop+1)+1, loop.set_pt[iLoop], szBuf_Loop);
  lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
  Real_to_Hexa(10*(iLoop+1)+3, loop.Kc[iLoop], szBuf_Loop);
  lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
  Real_to_Hexa(10*(iLoop+1)+5, loop.Ti[iLoop], szBuf_Loop);
  lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
//   Real_to_Hexa(10*(iLoop+1)+7, loop.Td[iLoop], szBuf_Loop);
//   lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
  Check_Sum(szWrite_Loop[iLoop]);
  fWrite_Only(szWrite_Loop[iLoop]);
}
void fSet_PD(int iLoop)
{
char szBuf_Loop[17];
  lstrcpy(szWrite_Loop[iLoop],":3A02\0");

// 74 characters with write mode
```

```
  Real_to_Hexa(10*(iLoop+1)+1, loop.set_pt[iLoop], szBuf_Loop);
  lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
  Real_to_Hexa(10*(iLoop+1)+3, loop.Kc[iLoop], szBuf_Loop);
  lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
//   Real_to_Hexa(10*(iLoop+1)+5, loop.Ti[iLoop], szBuf_Loop);
//   lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
  Real_to_Hexa(10*(iLoop+1)+7, loop.Td[iLoop], szBuf_Loop);
  lstrcat (szWrite_Loop[iLoop], szBuf_Loop);
  Check_Sum(szWrite_Loop[iLoop]);
}
////////////////////////////////////////////////////////////////////
//
// Change the sampling rate
//
////////////////////////////////////////////////////////////////////
void fChange_Ts(float T_Sample)
{
char szBuf[17], szChange_Rate[40];

 lstrcpy (szChange_Rate, ":1A02\0");
 Real_to_Hexa(19,T_Sample, szBuf);
 lstrcat (szChange_Rate, szBuf);
 Check_Sum(szChange_Rate);
 fWrite_Only(szChange_Rate);

}

////////////////////////////////////////////////////////////////////
//
//  Read IO variable
//
////////////////////////////////////////////////////////////////////
void fRead_IO_Var( char *szCommand, char *szRead_Buf1)
{
int iChar, nReadChars, nStatus;
  iChar = 0;
  nStatus = WriteComm (nComID,
  szCommand,
  lstrlen(szCommand));
  if (nStatus < 0)
  MessageBox (NULL,
  "Output Comm error.",
  "Message",
  MB_OK);
  else
  {
  lstrcpy(szRead_Buf1 ,"");
  do
  {
  rcomm1:
  nReadChars = ReadComm (nComID, cInBuf, 1);
  if (nReadChars != 1) goto rcomm1;
  if( isxdigit(cInBuf[0])||((cInBuf[0] == 58)

  || (cInBuf[0] == 59) ))
{
```

```
szRead_Buf1[iChar] = cInBuf[0];
iChar += 1 ;
}
  }
  while ( cInBuf[0] != 13 ) ;
  szRead_Buf1[iChar]  = '\0';
  }
}
////////////////////////////////////////////////
//
// Function for identification step
//
////////////////////////////////////////////////
void fIdent_Step (int i_step)
{
int step_sw, iMan_Loop_0, iMan_Loop_1, iMan_Loop_2,iKK, iValue;
float T_Sample;
char szBuf1[30], szBuf2[30];

  step_sw = i_step;
  lstrcpy(szBuf1, ":1A02\0");
switch(step_sw)
{
case 0:
  iAuto_Loop = 0;   // Auto
//   iMan_Loop_0 = 0;  // Manual mode
  iMan_Loop_1 = 1;
  iMan_Loop_2 = 2;
//   u_ss_0 = iTwx[4]; // Manual
  u_ss_1 = iTwx[5];
  u_ss_2 = iTwx[6];
  break;

// case 1:
//   iAuto_Loop = 0;
//   iMan_Loop_0 = 0;  //
//   iMan_Loop_2 = 2;
//   u_ss_0 = iTwx[4]; //
//   u_ss_1 = iTwx[5];
//   u_ss_2 = iTwx[6];
//   break;

    case 1:
  iAuto_Loop = 0;   // Auto
//   iMan_Loop_0 = 0;  // Manual mode
  iMan_Loop_1 = 1;
  iMan_Loop_2 = 2;
//   u_ss_0 = iTwx[4]; // Manual
  u_ss_1 = iTwx[5];
  u_ss_2 = iTwx[6];
  break;

case 2 :
  iAuto_Loop = 1;
  iMan_Loop_1 = 0;
  iMan_Loop_2 = 2;
```

```
  u_ss_1 = iTwx[4];
  u_ss_2 = iTwx[6];
  break;

case 3 :
  iAuto_Loop = 2;
  iMan_Loop_1 = 1;
  iMan_Loop_2 = 0;
  u_ss_1 = iTwx[5];
  u_ss_2 = iTwx[4];
  break;
  }
  fAuto_Mode(iAuto_Loop);    //  Auto
  loop.set_pt[iAuto_Loop] =                          //  Auto
  iTwx[iAuto_Loop]*0.048125 - 220.0 +
Delta_Set_Temp;  // Auto
  loop.Kc[iAuto_Loop] = Kc_i[iAuto_Loop];   // Auto
  loop.Ti[iAuto_Loop] = Ti_i[iAuto_Loop];   // Auto
  fSet_PI(iAuto_Loop);                      // Auto
//   fLoop_Bias_Write(iAuto_Loop,0.0);
//   T_Sample = Real_Sec;
//   fMan_Mode(iMan_Loop_0);   // Manual
  fMan_Mode(iMan_Loop_1);
  fMan_Mode(iMan_Loop_2);
//   fInt_to_Hexa_One(125+iAuto_Loop*10,0, szBuf2);  //
//   fWrite_Only(szBuf2);                            //
//   fInt_to_Hexa_One(123+iMan_Loop_0*10,u_ss_0+250, szBuf2);  // Manual
//   fWrite_Only(szBuf2);                                       // Manual

  fInt_to_Hexa_One(123+iMan_Loop_1*10,u_ss_1, szBuf2);
  fWrite_Only(szBuf2);
  fInt_to_Hexa_One(123+iMan_Loop_2*10,u_ss_2, szBuf2);
  fWrite_Only(szBuf2);
  fChange_Ts(0.2);
}
////////////////////////////////////////////////////////////////////
//
//   Function for Integer to Hexa conversion for 2 memory address
//   write
//
////////////////////////////////////////////////////////////////////
void fInt_to_Hexa(int VMem_No_1, int iValue_1,
int VMem_No_2, int iValue_2, char *szBuf)
{
wsprintf(szBuf, ":1A02%04X%04X%04X%04X\0",
 VMem_No_1 - 1, iValue_1,
 VMem_No_2 - 1, iValue_2);
Check_Sum(szBuf);
}
////////////////////////////////////////////////////////////////////
//
//   Function for Recursive Identification using Matlab routine
//
////////////////////////////////////////////////////////////////////
void fRecursive_Ident(char *szBuf4)
{
```

```
UINT      cblength;
double    du, dy_T[3], dzz[2], *thm_real;
int iKK, iLen3;
char      szBuf3[80];
  du = (iTwx[iAuto_Loop + 2] - u_ss)/ 32000.0;
  dzz[1]=du;
  for (iKK=0;iKK<3;++iKK)
 dy_T[iKK]=(iTwx[iKK]-y_ss_T[iKK])/10000.0;
  for (iKK=0;iKK<3;++iKK)
  {
 dzz[0]=dy_T[iKK];
//  memcpy((char *) mxGetPr(zz_vec), (char *) dzz,
//  2*sizeof(double));
//  mxSetName(zz_vec, "zz");
//  engPutMatrix(ep, zz_vec);
 switch (iKK)
 {
 case 0 :
  lstrcpy(szBuf3,"[thm_1, yhat_1,P_1,phi_1]\0");
  break ;
 case 1 :
  lstrcpy(szBuf3,"[thm_2, yhat_2,P_2,phi_2]\0");
  break ;
 case 2 :
  lstrcpy(szBuf3,"[thm_3, yhat_3,P_3,phi_3]\0");
  break ;
 }
 if (iCheck  == 0)
 {
 lstrcat(szBuf3," = rarx(zz , nn , 'ff' ,0.99 )\0");
//  MessageBox(NULL, szBuf3, "Message",MB_OK);
//
//  engEvalString(ep, szBuf3);
 switch (iKK)
 {
 case 0 :
//   thm = engGetMatrix (ep, "thm_1");
  break ;
 case 1 :
//   thm = engGetMatrix (ep, "thm_2");
  break ;
 case 2 :
//   thm = engGetMatrix (ep, "thm_3");
  break ;
 }
 }
 else
 {
 switch (iKK)
 {
 case 0 :
 lstrcat(szBuf3,

"=rarx(zz,nn,'ff',0.99,thm_1',P_1,phi_1)\0");
//  engEvalString(ep, szBuf3);
//  thm = engGetMatrix (ep, "thm_1");
```

```
 break ;
 case 1 :
 lstrcat(szBuf3,

"=rarx(zz,nn,'ff',0.99,thm_2',P_2,phi_2)\0");
//  engEvalString(ep, szBuf3);
//  thm = engGetMatrix (ep, "thm_2");
 break ;
 case 2 :
 lstrcat(szBuf3,

"=rarx(zz,nn,'ff',0.99,thm_3',P_3,phi_3)\0");
//  engEvalString(ep, szBuf3);
//  thm = engGetMatrix (ep, "thm_3");
 break ;
 }
 }
 lstrcpy(szBuf3, szBuf4);
//  thm_real = mxGetPr(thm);
//  iLen3 = sprintf(szBuf4,
//   "%5.1f %10.8f %10.8f \n",
//   Real_Time,
//   thm_real[0], thm_real[1]);
//     cblength = _lwrite(pFile, szBuf4,iLen3);
 if(cblength == (UINT)HFILE_ERROR)
 {
 MessageBox(NULL, "Error: Cannot write to file",
 szProgName, MB_OK | MB_ICONINFORMATION);
 return;
 }
 }  // end of for loop

 iCheck = 1;
}
////////////////////////////////////////////////////////////////////
//
// Random number addition to control input to process
//
////////////////////////////////////////////////////////////////////
void fAdd_Random(int iRand_Loop, double fMag_Noise)
{
//double  *drandom;                     // iAuto_Loop ; 0, 1, 2;
float   f_random;
int     u_random, u_random_neg, iLen; // fMag_Noise ; 0.0  --  0.10
char    szBuf[30];

 if ( fMag_Noise == 0.0 )
  {
  u_random = 0;
  u_random_neg = 0;
  }
 else if ( (fMag_Noise > 0.0) && (fMag_Noise <= 0.39) )
  {
//   engEvalString(ep, "arandom = rand(1)" );
//   prandom = engGetMatrix(ep, "arandom");
//   drandom = mxGetPr(prandom);
```

```
  f_random = ran3(&iSeed);
  if (f_random > 0.5)
{
u_random = u_ss * fMag_Noise * (f_random - 0.5);
u_random_neg = 0;
}
  else if (f_random <=0.5)
{
u_random_neg = u_ss * fMag_Noise *(0.5 - f_random);
u_random = 0;
}
  }
 else if ( fMag_Noise > 0.390)
  _exit(0);

  fInt_to_Hexa((21+iRand_Loop)*10 + 1, u_random,
(21+iRand_Loop)*10 + 2,
u_random_neg,
szBuf);
  fWrite_Only(szBuf);
  u_random_sum = u_random - u_random_neg;
}
////////////////////////////////////////////////////////////////////
//
//    Set all the set point to same and write to plc loop
//
////////////////////////////////////////////////////////////////////
void fSet_All_Setpt(void)
{
int iLoop;
char szBuf_Loop[17];
for (iLoop = 0; iLoop < 3; ++iLoop)
{
lstrcpy(szWrite_Loop[iLoop],":1A02\0");
loop.set_pt[iLoop]=Set_Temp[iLoop];
Real_to_Hexa(10*(iLoop+1)+1, Set_Temp[iLoop], szBuf_Loop);
lstrcat(szWrite_Loop[iLoop], szBuf_Loop);
Check_Sum(szWrite_Loop[iLoop]);
fWrite_Only(szWrite_Loop[iLoop]);
 }
}
////////////////////////////////////////////////////////////////////
//
//   Returns a uniform random deviate between 0.0 and 1.0. Set idum to
//  any negative value to initialize or reinitialize the sequence*/
//  According to Knuth, any large MBIG, and any smaller (but still
//  large) MSEED can be substituted for the above values
//
////////////////////////////////////////////////////////////////////
float ran3(long *idum)
{
 static int inext, inextp;
 static long ma[56];
    static int iff=0;
    long mj,mk;
    int i,ii,k;
```

```
    if (*idum < 0 || iff == 0) {             /* Initialization*/
        iff=1;
        mj=MSEED-(*idum < 0 ? -*idum : *idum);
        mj %= MBIG;
        ma[55]=mj;
        mk=1;
        for (i=1;i<=54;i++) {
            ii=(21*i) % 55;
            ma[ii]=mk;
            mk=mj-mk;
            if (mk < MZ) mk += MBIG;
            mj=ma[ii];
  }
        for (k=1;k<=4;k++)
            for (i=1;i<=55;i++) {
                ma[i] -= ma[1+(i+30) % 55];
                if (ma[i] < MZ) ma[i] += MBIG;
      }
        inext=0;
        inextp=31;
        *idum=1;
    }

/*  Here is where we start, except on initialization.*/
    if (++inext == 56) inext=1;
    if (++inextp == 56) inextp=1;
 mj=ma[inext]-ma[inextp];
    if (mj < MZ) mj += MBIG;
    ma[inext]=mj;
    return mj*FAC;
}
////////////////////////////////////////////////////////////////////////
//
//   Function for Integer to Hexa conversion for 1 memory address
//
////////////////////////////////////////////////////////////////////////
void fInt_to_Hexa_One(int VMem_No, int iValue, char *szBuf)
{
wsprintf(szBuf, ":1202%04X%04X\0",
 VMem_No - 1, iValue);
Check_Sum(szBuf);
}
////////////////////////////////////////////////////////////////////////
//
//  Function for PLC Program Mode
//
////////////////////////////////////////////////////////////////////////
void fProg_Mode(void)
{
char szBuf[15];
lstrcpy(szBuf,":0E330000F1CD;\0");
fWrite_Only(szBuf);
}
////////////////////////////////////////////////////////////////////////
//
```

```
//  Function for PLC Run Mode
//
////////////////////////////////////////////////////////////////////
void fRun_Mode(void)
{
char szBuf[11];
lstrcpy(szBuf, ":0A32F5CE;\0");
fWrite_Only(szBuf);
}
////////////////////////////////////////////////////////////////////
//
//  Function for  loop enable
//
////////////////////////////////////////////////////////////////////
void fLoop_Enable(int iLoop_No)
{
char szBuf[15], szBufOne[1];
lstrcpy(szBuf, ":0E5D800\0");
// itoa(iLoop_No, szBufOne,10)
lstrcat(szBuf, itoa(iLoop_No, szBufOne,10));
Check_Sum(szBuf);
fWrite_Only(szBuf);
}
////////////////////////////////////////////////////////////////////
//
//  Function for  loop disable
//
////////////////////////////////////////////////////////////////////
void fLoop_Disable(int iLoop_No)
{
char szBuf[15], szBufOne[1];
lstrcpy(szBuf, ":0E5E000\0");
lstrcat(szBuf,itoa(iLoop_No, szBufOne,10));
Check_Sum(szBuf);
fWrite_Only(szBuf);
}
////////////////////////////////////////////////////////////////////
//
//  Function for loop bias value write
//
////////////////////////////////////////////////////////////////////
void fLoop_Bias_Write(int iLoop_No, float fValue)
{
char szBuf_1[24];
char szBuf_2[10];
number temp;
temp.x = fValue;
switch(iLoop_No)
  {
  case 0 :
  lstrcpy(szBuf_1, ":16600131\0");
  break ;
  case 1 :
  lstrcpy(szBuf_1, ":16600231\0");
  break ;
  case 2 :
```

```
  lstrcpy(szBuf_1, ":16600331\0");
  break ;
  }
wsprintf (szBuf_2, "%04X%04X\0", temp.n[1],temp.n[0]);
lstrcat (szBuf_1, szBuf_2);
Check_Sum(szBuf_1);
fWrite_Only(szBuf_1);

}
```

Bibliography

[1] Advanced Energy Industries, Inc., Fort Collins, Colorado. *User Manual for the RFX 600 Fenerator*, 1990.

[2] H. W. Anderson, M. Kummel, and S. B. Jorgensen. Dynamics and identification of a binary distillation column. *Chem. Eng. Sci.*, 44:2571–2581, 1989.

[3] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *J. Robotic Systems*, 1(2):123–140, 1984.

[4] K. J. Åström and T. Hägglund. Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5):645–651, 1984.

[5] K. S. Balakrishnan, W. Cho, and T. F. Edgar. Comparison of controller tuning methods for temperature uniformity control in a rapid thermal processor. In *SPIE*, volume 2336, pages 71–75, 1994.

[6] G. Bentini, L. Correra, and C. Donolato. Defects introduced in silicon wafers during rapid isotermal annealing: Thermoelastic and thermoplastic effects. *J. Appl. Phys.*, 56(10):2922–2929, Nov 1984.

[7] R. Byron Bird, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena*. John Wiley & Sons, Inc., New York, 1960.

[8] R. D. Braatz and M. Morari. Minimizing the euclidean condition number. *SIAM J. Control Optimization*, 32:1763–1768, 1994.

[9] T. Breedijk, T. F. Edgar, and I. Trachtenberg. Parameter estimation and nonlinear predictive control for RTP. In *ADCHEM '94*, pages 85–91, 1994.

[10] Terence Breedijk. *Model Identification and Nonlinear Predictive Control of Rapid Thermal Processing System.* PhD thesis, University of Texas at Austin, 1994.

[11] S. A. Campbell, K. H. Ahn, K. L. Knutson, B. Y. H. Liu, and J. D. Leighton. Steady-state thermal uniformity and gas flow patterns in a rapid thermal processing chamber. *IEEE Trans. on Semicon. Man.*, 4(1):14–20, 1991.

[12] S. Chatterjee, T. F. Edgar, and I. Trachtenberg. Modeling and control of RTCVD of polysilicon. *First International Rapid Thermal Processing Conference, Scottsdale,AZ*, 1993.

[13] J. Chen, J. S. Freudenberg, and C. N. Nett. The role of the condition number and the relative gain array in robust analysis. *Automatica*, 30:1029–1035, 1994.

[14] J. Chen and C. N. Nett. Bounds on generalized structured singular values via the perron root of matrix majorants. *Systems Control Lett.*, 19:439–449, 1992.

[15] K.-C. Chen, H.-H. Shih, C.-C. Hsueh Y.-L.Hwang, H. Chung, S. Pan, and C.-Y. Lu. Applications of single-wafer rapid thermal processing to the manufac-

ture of advanced flash memory. *IEEE Trans. Semiconductor Manufacturing*, 16(2):128–137, May 2003.

[16] M.-S. Chiu and Y. Arkun. A methodology for sequential design of robust decentralized control systems. *Automatica*, 28(5):997–1001, 1992.

[17] W. Cho, Balakrishnan K. S., T. F. Edgar, and I. Trachtenberg. Control of remote plasma-enhanced chemical vapor deposition of silicon nitride. In *5th Int.Sym.on Process Systems Engineering*, pages 1235–1241, 1994.

[18] Y. M. Cho and T. Kailath. Model identification in rapid thermal processing systems. *IEEE Trans. Semicond. Manufact.*, 6(3):233–245, August 1993.

[19] J. Y. Choi and H. M. Do. A learning approach of wafer temperature control in a rapid thermal processing system. *IEEE Transactions on Semiconductor Manufacturing*, 14(1):1–10, 2001.

[20] C. Davis, M. Moslehi, A. Bowling, and J. D. Luttmer. Microelectronics Manufacturing Science and Technology: Equipment and Sensor Technologies. *TI Technical J.*, pages 20–43, Sep.-Oct. 1992.

[21] C. A. Desoer and Y. T. Wang. On the generalized nyquist stability criterion. *IEEE Trans. Automatic Control*, AC-25(2):187–196, APRIL 1980.

[22] D. P. DeWitt and G. D. Nutter. *Theory and Practice of Radiation Thermometry*. Wiley, New York, 1988.

[23] J. Doyle. Analysis of feedback systems with structured undertainties. *IEE Proc.*, 129 pt.D(6):242–250, Nov 1982.

[24] T. F. Edgar, S. W. Butler, W. J. Campbell, C. Pfeiffer, C. Bode, S. B. Hwang, K. S. Balakrishnan, and J. Hahn. Automatic control in microelectronics manufacturing:practices, challenges, and possibilities. *Automatica*, 36:1567–1603, 2000.

[25] Kiefer J. Elliott. Personal communication.

[26] M. K. H. Fan and A. L. Tits. m-form numerical range and computation of the structured singular value. *IEEE Trans. Auto. Contr.*, 33(3):284–289, Mar 1988.

[27] Alejandro L. Garcia. *Nemerical Methods for Physics.* Prentice Hall, 2nd edition, 2000.

[28] C. E. Garcia and A. M. Morshedi. Quadratic programming solution of dynamic matrix control (qdmc). *Chem.Eng.Commun.*, 46:73–87, 1986.

[29] M. Gevers. Connecting identification and robust control. In *9th IFAC Symp. on Identification and System Parameter Estimation*, volume 31, pages 855–860, 1991.

[30] P. W. Gofton. *Mastering Serial Communications.* Sybex, Alemeda, CA, 1986.

[31] G. H. Golub and C. F. Van Loan. *Matrix computations.* The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[32] M. Goodwin. *Serial Communications Programming in C and C++.* MIS:Press, New York, 1992.

[33] P. Grosdidier. Analysis of interaction direction with the singular value decomposition. *Computers chem Engg*, 14(6):687–689, 1990.

[34] P. Grosdidier and M. Morari. Interaction measures for systems under decentralized control. *Automatica*, 22(3):309–319, 1986.

[35] P. Grosdidier, M. Morari, and B. R. Holt. Closed-loop properties from steady-state gain information. *Ind. Eng. Chem. Fundam.*, 24:221–235, 1985.

[36] Rosenbrock H. H., editor. *Computer-aided control system design*. Academic Press, London, 1974.

[37] G. Held. *Understanding Data Communications*. SAMS, 3rd edition, 1991.

[38] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[39] M. Hovd and S. Skogestad. Simple frequency-dependent tools for control system analysis, structure selection and design. *Automatica*, 28:989–996, 1992.

[40] S.-Y. Hsu, T.-Y.Wang, H.-H Shih, K.-C.Chen, Y.-L.Hwang, C.-C. Hsueh, H. Chung, S. Pan, and C.-Y.Lu. Applications of single-wafer thermal processing to 0.15 μm floating gate flash memory. *IEEE Trans. Semiconductor Manufacturing*, 16:147–154, May 2003.

[41] S. B. Hwang. *Temperature Control and Characterization of Silicon-Germanium Growth by Rapid Thermal Chemical Vapor Deposition*. PhD thesis, University of Texas at Austin, 2002.

[42] Postlethwaite I. and Y. K. Foo. Robustness with simultaneous pole and zeor movement across the $j\omega$ axis. *Automatica*, 21:433–443, 1985.

[43] Yaw-Kuen Jan and Ching-An Lin. Lamp configuration design for rapid thermal processing systems. *IEEE Transactions on Semiconductor manufacturing*, 11(1):75–84, February 1998.

[44] C. W. Koung and J. F. MacGregor. Design of identification experiments for robust control. a geometric approach for bivariate processes. *Ind. Chem. Eng. Res.*, 32:1658–1666, 1993.

[45] C. W. Koung and J. F. MacGregor. Identification for robust multivariable control:the design of experiments. *Automatica*, 30:1541 – 1554, 1994.

[46] H. Lau, J. Alvarez, and K. F. Jensen. Synthesis of control structures by singular value analysis: dynamic measures of sensitivity and interaction. *AIChE J.*, 31(3):427–439, Mar 1985.

[47] J. Lee, W. Cho, and T. F. Edgar. Effects of diagonal input uncertainties and element uncertainties in ill-conditioned processes. *Ind.Eng.Chem.Res*, 37:1009 – 1017, 1998.

[48] J. Lee, W. Cho, and T. F. Edgar. Iterative identification methods for ill-conditioned processes. *Ind. Eng. Chem. Res.*, 37:1018–1023, 1998.

[49] J. Lee and J. Y. Choi. Design of multiloop controller. *J. KIChE*, 31:272–278, 1993.

[50] J. T. Lee, W. Cho, and T. F. Edgar. Multiloop PI controller tuning for interacting multivariable processes. *Computers Chem. Engng*, 22(11):1711–1723, 1998.

[51] K.S. Lee, S. H. Bang, and K. S. Chang. Feedback-assisted iterative learning control based on an inverse process model. *J. of Process Control*, 4(2):77–89, May 1994.

[52] W. Li and J. H. Lee. Control relevant identification of ill-conditioned systems: Estimation of gain directionality. *Computers and Chem. Eng.*, 20(8):1023–1042, 1996.

[53] S. Lin and H.-S. Chu. Thermal uniformity of 12-in silicon wafer in linearly ramped-temperature transient rapid thermal processing. *IEEE Trans. on Semicon. Manufacturing*, 14(2):143–151, 2001.

[54] A. P. Loh, C. C. Hang, C. K. Quek, and V. U. Vasnani. Autotuning of multiloop proportional-integral controllers using relay feedback. *Ind. Engng. Chem. Res.*, 32:1102–1107, 1993.

[55] H. A. Lord. Thermal and stress analysis of semiconductor wafers in a rapid thermal processing oven. *IEEE Trans.on Semicon. Man.*, 1:105–114, Aug 1988.

[56] M. Loukides. *UNIX for FORTRAN Programmers*. O'Reilly & Associates, INC., Sebastopol, CA, 1990.

[57] W. L. Luyben. Simple method for tuning SISO controllers in multivariable systems. *Ind. Eng. Chem. Process Des. Dev.*, 25:654–660, 1986.

[58] J. M. Maciejowski. *Multivariable Feedback Design.* Addison Wesley, New York, 1989.

[59] D. Q. Mayne. The design of linear multivariable systems. *Automatica*, 9:201–207, 1976.

[60] K.A. McDonald, A. Palazoglu, and B. W. Bequette. Impact of model uncertainty descriptions for high-purity distillation control. *AIChE J.*, '35:1996–2004, 1989.

[61] M. Morari and E. Zafiriou. *Robust Process Control.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[62] M. M. Moslehi, C. Davis, and A. Bowling. Microelectronics manufacturing science and technology:single-wafer thermal processing and wafer cleaning. *TI Technical J.*, pages 44–64, Sep.-Oct. 1992.

[63] C. N. Nett and V. Manousiouthakis. Euclidean condition and block relative gain:connentions, conjectures and clarifications. *IEEE Trans. Auto. Contr.*, AC-32(5):405–407, May 1987.

[64] B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling and Control.* Oxford University Press, 3rd edition, 1994.

[65] H. M. Park, T. Y. Yoon, and O. Y. Kim. Optimal control of rapid thermal procesing systems by empirical reduction of modes. *Ind. Eng. Chem. Res.*, 38:3964–3975, 1999.

[66] I. Postlethwaite and MacFarlane J. M. *A complex variable approach to the analysis of linear multivariable feedback systems.* Springer, Berlin, 1979.

[67] S. J. Qin, G. W. Scheid, and T. J. Reid. Adaptive run-to-run control and monitoring for a rapid thermal processor. *J.Vac.Sci. Technol. B*, 21:301–310, Jan/Feb 2003.

[68] D. E. Rivera, M. Morari, and S. Skogestad. Internal model control, 4. PID controller design. *Ind. Eng. Process Design Dev.*, 25:252–260, 1986.

[69] F. Roozeboom. Temperature control and system design aspects in rapid thermal processin. In *Mat. Res. Soc. Symp.*, volume 224. Materials Research Society, 1991.

[70] F. Roozeboom. Manufacturing equipment issues in rapid thermal processing. In R. B. Fair, editor, *Rapid Thermal Processing Science and Technology*, chapter 9, pages 349–423. Academic Press, Boston, 1993.

[71] F. Roozeboom and N. Parekh. Rapid thermal processing systems:a review with emphasis on temperature control. *J.Vac.Sci.Technology.B*, 8:1249–1259, Nov/Dec 1990.

[72] Tsutomu Sato. Spectral emissivity of silicon. *Japanese Journal of Applied Physics*, 6:339–347, March 1967.

[73] C. D. Schaper, Y. Cho, P. Park, S. Norman, P. Gyugyi, G. Hoffmannand, S. Balemi, S. Boyd, G. Franklin, T. Kailath, and S. Saraswat. Modeling and control of rapid thermal processing. *SPIE*, 1595:2–17, 1991.

[74] C. D. Schaper, Y. M. Cho, and T. Kailath. Low-order modeling and dynamic characterization of rapid thermal processing. *Appl.Phys.A*, 54:317–326, 1992.

[75] C. D. Schaper, T. Kailath, and Y. J. Lee. Decentralized control of wafer temperature for multizone rapid thermal processing systems. *IEEE Trans. on Semiconductor Manufacturing*, 12(2):193–199, May 1999.

[76] C. D. Schaper, M. M. Moslehi, K. C. Saraswat, and T. Kailath. Modeling, identification, and control of rapid thermal processing systems. *J. Electrochem. Soc.*, 141(11):3200–3209, November 1994.

[77] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp. *Process Dynamics and Control*. John Wiley & Sons, New York, 2004.

[78] D. E. Seborg, T. F. Edgar, and S. L. Shah. Adaptive control strategies for process control:a survey. *AIChE Journal*, pages 881–913, 1986.

[79] SensArray Corporation, 2388 Walsh Avenue, Santa Clara, CA 95051. *Using SensArray Temperature Instrumented Wafers*.

[80] S.-H. Shen and C.-C. Yu. Use of relay-feedback test for automatic tuning of multivariable systems. *AIChE J.*, 40(4):627–646, Apr 1994.

[81] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. McGraw Hill, Washington, 2nd edition, 1981.

[82] Siemnes Industrial Automation, Inc., Johnson City, TN. *SIMATIC TI 500/505 TISOFT 2 Release 4.2*, 3rd edition, 1992.

[83] R. Singh and J. Nulman. Development trends in the direction of rapid isothermal processing (RIP) dominated silicon integrated circuit fabrication. In *Materials Issues in Silicon Integrated Circuit Processing*, volume 71. Materials Research Society, 1991.

[84] S. Skogestad and M. Morari. Effect of disturbance directions on closed-loop performance. *Ind. Eng. Chem. Res.*, 26:2029–2035, 1987.

[85] S. Skogestad and M. Morari. Implication of large elements on control performance. *Ind. Eng. Chem. Res.*, 26:2323–2330, 1987.

[86] S. Skogestad and M. Morari. Some new properties of the structure singular value. *IEEE Trans. AC*, 33:1151–1154, Dec 1988.

[87] S. Skogestad and M. Morari. Robust performance of decentralized control systems by independent designs. *Automatica*, 25(1):119–125, 1989.

[88] S. Skogestad, M. Morari, and J. C. Doyle. Robust control of ill-conditioned plants:high-purity distillation. *IEEE Trans. Auto.Contr.*, 33(12):1092–1105, Dec. 1988.

[89] C. L. Smith. *Digital Computer Process Control.* Intext Educational Publishers, New York, 1972.

[90] G. Stanley, M. Marino-Galarraga, and T. J. McAvoy. Shortcut operability analysis i: the relative disturbance gain. *Ind. Eng. Chem. Process Des. Dev.*, 24:1181–1188, 1985.

[91] J. D. Stuber, T. F. Edgar, and T. Breedijk. Model based control of rapid thermal processes. *Proc. Electrochem Soc.*, 95(4):113–122, 1995.

[92] Texas Instruments, Johnson City, Tennesse. *Series 505 Discrete Input Module User's Manual*, 1987.

[93] Texas Instruments, Johnson City, Tennesse. *Series 505 I/O Simulators User's Manual*, 1987.

[94] Texas Instruments, Johnson City, Tennesse. *Series 505 Relay Output Module User's Manual*, 1987.

[95] Texas Instruments, Johnson City, Tennesse. *TIWAT I Series 505 Network Interface User's Manual*, 1988.

[96] Texas Instruments, Johnson City, Tennesse. *505 High Speed Counter and Encoder Module User Manual*, 1989.

[97] Texas Instruments, Johnson City, Tennesse. *Series 500/505 Programming Reference Manual*, 2nd edition, 1990.

[98] Texas Instruments, Johnson City, Tennessee. *Series 500/505 Programming Reference Manual*, 2nd edition, March 1990.

[99] Texas Instruments, Johnson City, Tennesse. *Model 505-7012 Eight-input Four-output Analog Module User Manual*, 1991.

[100] Texas Instruments, Johnson City, Tennesse. *Series 505 386/ATM Coprocessor User Manual*, 1991.

[101] Texas Instruments, Inc., Johnson City, Tennessee. *Non-intelligent Terminal Protocol*, 1980.

[102] Texas Instruments, Inc., Johnson City, Tennessee. *System Specification, Transparent Byte Protocol*, 1987.

[103] P. Vandenabeele and K. Maex. Influence of temperature and backside roughness on the emissivity of Si wafers during rapid thermal processing. *J. Appl. Phys*, 72:5867–5875, 1992.

[104] R. A. Weimer, D. M. Eppich, K. L. Beaman, D. C. Powell, and F. González. Contrasting single-wafer and batch processing for memory devices. *IEEE Trans. Semiconductor Manufacturing*, pages 138–146, May 2003.

[105] D. R. Yang, K. S. Lee, H. J. Ahn, and J. H. Lee. Experimental application of a quadratic optimal iterative learning control method for control of wafer temperature uniformity in rapid thermal processing. *IEEE Transactions on Semiconductor Manufacturing*, 16(1):36–44, 2003.

[106] C. C. Yu and W. L. Luyben. Robustness with respect to integral controllability. *Ind. Eng. Chem. Res.*, 26:1043–1045, 1987.

Vita

Wonhui Cho, son of Kwang-Joon Cho and Keum-Hee Kim, was born in Seoul, Korea, on July 7, 1957. After graduating from Seoul National University in February of 1980 with a Bachelor of Science degree in Chemical Engineering, he entered the Department of Chemical Engineering in Korea Advanced Institute of Science and Technology in March, 1981 and graduated with a Master of Science degree in Chemical Engineering in February, 1983. Between January of 1983 and May of 1986, he worked at Cheil Sugar & Co.(a branch of Samsung at that time) as a programmer and system analyst. He entered the Department of Physics at The Texas A & M University in August of 1986 and graduated with a Master of Science in Physics in August of 1988. He was a graduate student at the Physics department at The University of Texas between September of 1988 and December of 1989. In January of 1990 he transferred to the Department of Chemical Engineering at The University of Texas for Ph.D. study. He has worked for Momentum Technical Consulting in Austin, Texas between 1997 and 1999 and later worked for Advanced Micro Devices, Inc. between 2000 and 2002. In July of 1986 he was married to Hyung Joo Chin. They have lovely children, Sherry Ann and Peter Stanley, who were born in College Station, and Austin Texas, respectively.

Permanent address: 1330 Roadrunner Drive
Cedar Park, Texas 78613

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.